

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Mitja Roblek

Namizna aplikacija za beleženje tehtanj tovornih vozil

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2014

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Mitja Roblek

Namizna aplikacija za beleženje tehtanj tovornih vozil

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Matjaž Kukar

Ljubljana, 2014

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Kandidat naj razvije namizno aplikacijo namenjeno tehtanju tovornih vozil z industrijskimi tehtnicami. Razišče naj primerne načine in protokole za strojno in programsko povezavo tehtnic z računalnikom in okrog njih razvije aplikacijo, ki zadošča uporabniškim zahtevam. Posebno pozornost naj posveti primerni arhitekturi in organizaciji hranjenja podatkov. Aplikacijo naj testira v dejanskem produkcijskem okolju in z dejanskimi uporabniki, ter poda oceno uspešnosti implementacije v praksi.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Mitja Roblek, z vpisno številko **63050347**, sem avtor diplomskega dela z naslovom:

Namizna aplikacija za beleženje tehtanj tovornih vozil

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Matjaža Kukarja,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 10. september 2014

Podpis avtorja:

Zahvaljujem se doc. dr. Matjažu Kukarju za strokovno pomoč in vodenje pri izdelavi diplomskega dela. Hvala tudi sodelavcem v podjetju AHATServis, d.o.o., ter vsem prijateljem, ki so kakorkoli pomagali k nastanku tega dela. Posebna zahvala pa gre moji družini, ki mi je omogočila študij in me vseskozi podpirala.

Kazalo

Povzetek

Abstract

| | | |
|-------------------|---|-----------|
| Poglavje 1 | Uvod | 1 |
| Poglavje 2 | Uporabljena orodja | 3 |
| 2.1 | Strojna oprema | 3 |
| 2.2 | Programska orodja in tehnologije | 5 |
| 2.2.1 | Microsoft Visual Studio 2010 | 5 |
| 2.2.2 | MySQL Workbench | 6 |
| 2.2.3 | MySQL | 7 |
| 2.2.4 | Programski jezik C# | 9 |
| 2.2.5 | .NET Framework | 9 |
| Poglavje 3 | Komunikacija s tehtnico | 11 |
| 3.1 | Standard RS-232 | 11 |
| 3.1.1 | Napetosti logičnih nivojev | 12 |
| 3.1.2 | Signali RS-232 | 12 |
| 3.1.3 | Null modem | 14 |
| 3.1.4 | Okvir RS-232 | 17 |
| 3.1.5 | Napake pri prenosu | 19 |
| 3.2 | Komunikacija med tehtnico in računalnikom | 20 |
| 3.2.1 | Serijski vmesnik tehtnice Ohaus | 20 |
| 3.2.2 | Metode komuniciranja | 27 |
| Poglavje 4 | Izdelava aplikacije | 29 |
| 4.1 | Opis aplikacije | 29 |
| 4.2 | Uporabniške zahteve | 30 |

| | | |
|---|--|-----------|
| 4.2.1 | Primer uporabe | 32 |
| 4.3 | Podatkovna baza..... | 35 |
| 4.3.1 | Podatkovni model | 35 |
| 4.3.2 | Dostop do podatkovne baze | 39 |
| 4.4 | Uporabniški vmesnik..... | 45 |
| 4.5 | Poizvedbe za pregled statistike..... | 47 |
| Poglavje 5 Predstavitev in uporaba aplikacije kot jo vidi uporabnik..... | | 51 |
| 5.1 | Prijava v aplikacijo | 51 |
| 5.2 | Podjetje..... | 52 |
| 5.2.1 | Vnos novega podjetja..... | 52 |
| 5.2.2 | Spreminjanje podatkov obstoječega podjetja..... | 53 |
| 5.2.3 | Brisanje podjetja | 53 |
| 5.2.4 | Izvoz v CSV in tiskanje poročil | 53 |
| 5.3 | Material | 54 |
| 5.3.1 | Vnos novega materiala..... | 54 |
| 5.3.2 | Spreminjanje podatkov obstoječega materiala..... | 55 |
| 5.3.3 | Brisanje materiala | 55 |
| 5.3.4 | Izvoz v CSV in tiskanje poročil | 55 |
| 5.4 | Vozila | 56 |
| 5.4.1 | Vnos novega vozila..... | 56 |
| 5.4.2 | Spreminjanje podatkov obstoječega vozila..... | 56 |
| 5.4.3 | Brisanje vozila..... | 57 |
| 5.4.4 | Izvoz v CSV in tiskanje poročil | 57 |
| 5.5 | Tehtanja..... | 57 |
| 5.6 | Izvajanje tehtanj | 58 |
| 5.6.1 | Vnos (izbor) podatkov | 58 |
| 5.6.2 | Brisanje oziroma odznačitev vnesenih podatkov..... | 60 |
| 5.6.3 | Izvedba tehtanja | 60 |
| 5.6.4 | Tehtalni list | 62 |

| | | |
|-------------------|--|-----------|
| 5.7 | Nastavitve | 64 |
| 5.7.1 | Uporabniki | 65 |
| 5.7.2 | Tehtnica | 67 |
| 5.8 | Statistika..... | 68 |
| 5.8.1 | Po materialih..... | 68 |
| 5.8.2 | Po podjetjih..... | 70 |
| 5.9 | Grafična ponazoritev izvajanja tehtanja za voznika | 71 |
| Poglavje 6 | Sklepne ugotovitve..... | 73 |

Seznam uporabljenih kratic

| kratica | angleško | slovensko |
|----------------|------------------------------------|--|
| RS-232 | Recommended Standard 232 | priporočen standard 232 |
| USB | Universal Serial Bus | univerzalno serijsko vodilo |
| C# | C sharp | objektno usmerjen programski jezik |
| CSS | Cascading Style Sheets | kaskadne stilske predloge |
| HTML | HyperText Markup Language | jezik za označevanje nadbesedila |
| XML | Extensible Markup Language | razširljiv označevalni jezik |
| SQL | Structured Query Language | strukturirani povpraševalni jezik |
| MySQL | My Structured Query Language | odprtokodni sistem za upravljanje zbirk podatkov |
| ADO.NET | ActiveX Data Object for NET | zbirka programskih komponent za dostop do podatkov in podatkovnih servisov |
| DTE | Data Terminal Equipment | podatkovna terminalna oprema |
| DCE | Data Circuit-Terminating Equipment | podatkovna komunikacijska oprema |
| TD | Transmit Data | pošiljanje podatkov |
| RD | Receive Data | sprejemanje podatkov |
| GND | Common ground | signalna ozemljitev |
| RTS | Request To Send | zahteva po pošiljanju |
| CTS | Clear To Send | prosta za pošiljanje |
| DTR | Data Terminal Ready | podatkovni termin pripravljen |
| DSR | Data Set Ready | podatki pripravljeni za pošiljanje |
| LSB | Least Significant Bit | najmanj pomemben bit |
| MSB | Most Significant Bit | najbolj pomemben bit |

| | | |
|--------------|--|---|
| ASCII | American Standard Code for Information Interchange | ameriški standardni nabor za izmenjavo informacij |
| CR | Carriage Return | znak za konec vrstice |
| LF | Line Feed | znak za začetek nove vrstice |
| CSV | Comma Separated Values | datoteka s končnico .csv |

Povzetek

V sklopu diplomskega dela je bila izdelana namizna aplikacija ATS, ki je namenjena podjetjem za beleženje vhodnih in izhodnih tehtanj tovornih vozil. V uvodnem teoretičnem delu diplomskega dela so predstavljene in na kratko opisane uporabljene tehnologije in orodja za razvoj aplikacije ter sama strojna oprema, ki smo jo uporabili pri izdelavi in testiranju aplikacije. Aplikacija je bila zgrajena v programskem jeziku C# na podlagi .NET platforme in uporablja sistem za upravljanje s podatkovnimi bazami MySQL. Glavni praktični del diplomskega dela opisuje izdelavo podatkovne baze, uporabniškega vmesnika in natančen opis funkcionalnosti aplikacije na podlagi uporabniških zahtev. Za izdelavo aplikacije smo uporabili razvojno okolje Microsoft Visual Studio. Za prikaz strukture podatkovne baze pa smo uporabili orodje Microsoft Workbench, iz katerega smo izvozili entitetno-relacijski diagram obstoječe podatkovne baze.

Ključne besede: C#, namizna aplikacija, .NET, serijski vmesnik, MySQL, standard RS-232

Abstract

As a part of the diploma thesis, we developed the desktop application ATS, which is intended for recording the weight of goods vehicles at their arrival at a company and at their departure. The theoretical introduction of the thesis presents and shortly describes the technology and tools, which were used to develop the application and it also describes the hardware, which was used not only for development but also for testing. The application was built in the C# programming language, based on .NET framework and it uses MySQL, a relational database management system. The main part, which focuses on practice, describes the making of the database, the user interface and a detailed description of the application's functionality, based on user requirements. Microsoft Visual Studio served as the development environment for the application. To show the structure of the database, we used Microsoft Workbench tool, out of which we exported an entity-relationship diagram of the existing database.

Key words: C#, desktop application, .NET, serial interface, MySQL, standard RS-232

Poglavje 1 **Uvod**

Zakaj je en meter dolg ravno en meter in zakaj en kilogram tehta točno en kilogram, so vprašanja, ki se porodijo v zavednih otroških glavah, medtem ko odrasli urejenost tega sveta pogosto sprejemajo samoumevno. Tehtanje s svojim merilnim instrumentom – tehtnico – je staro toliko kot človeštvo. Tehtnica je skozi stoletja in tisočletja spremljala človeštvo kot temelj za enakovrednost in pravico, modrost in enotnost. Najdemo jo ne le kot predmet, temveč je s svojim simbolnim pomenom božje pravičnosti vmešana že v prostore religije. Močno pa je povezana z industrijsko dejavnostjo, ki je zelo razširjena v svetovnem gospodarstvu.

Tehtanje tovornih vozil (tovornjakov, vlakov ipd.) v industriji predstavlja zelo pomemben segment v preverjanju količin v transportu in logistiki. Lahko rečemo, da vsako večje podjetje premore cestno tehtnico ustrezne nosilnosti in velikosti, s pomočjo katere preverja oziroma nadzoruje tako prejete kot tudi oddane količine gotovega blaga, polproizvodov, proizvodov ter surovin, ki jih podjetja potrebujejo za svojo dejavnost.

Zaposlen sem v podjetju, ki je specializirano za prodajo cestnih tehtnic, zato sem prišel do ideje, da bi lahko našim strankam v sklopu nakupa cestne tehtnice ponudili programsko rešitev v obliki namizne aplikacije za beleženje mase vhodnih in izhodnih tehtanj tovornih vozil. Na trgu je prisotnih kar nekaj ponudnikov in programskih rešitev, vendar je zaradi specifičnosti dejavnosti med ponudniki zelo malo slovenskih podjetij. Programske rešitve, ki jih ponujajo tuji proizvajalci cestnih tehtnic (Dini Argeo [1], Leon Engineering [2] in Ohaus [3]), so zelo drage, nepregledne, vsebujejo neuporabne funkcionalnosti in imajo uporabniške vmesnike, ki niso prevedeni v slovenski jezik. Vse omenjene slabosti, ki smo jih ugotovili pri pregledu aplikacij za beleženje mase vhodnih in izhodnih tehtanj tovornih vozil, bomo poskušali z izdelavo lastne aplikacije odpraviti oziroma izboljšati. Naša aplikacija bo tako prosto dostopna, enostavna za uporabo, vsebovala bo le funkcionalnosti, ki so potrebne, in bo imela uporabniški vmesnik v slovenskem jeziku. Ker bomo aplikacijo izdelali sami, jo bomo lahko poljubno prilagajali potrebam stranke. Omenjene prednosti so pripomogle k odločitvi, da so v podjetju mojo idejo sprejeli in tako je prišlo do izdelave aplikacije, ki smo jo nato uporabili za diplomsko delo.

Namen diplomskega dela je prikaz in izdelava aplikacije, ki bo podjetjem omogočala beleženje vhodnih in izhodnih tehtanj tovornih vozil. To pomeni, da vozilo ob prihodu v podjetje stehtamo, kar predstavlja vhodno tehtanje. Pred izhodom iz podjetja vozilo ponovno stehtamo, kar predstavlja izhodno tehtanje. Ko je opravljeno vhodno in izhodno tehtanje, je tehtanje zaključeno. Na tehtalnem listu se potem izpišejo vsi podatki tehtanja, pri čemer se avtomatsko določi, ali je vozilo pripeljalo material (masa vhodnega tehtanja je večja od mase izhodnega tehtanja) ali odpeljalo material (masa vhodnega tehtanja je manjša od mase izhodnega tehtanja).

Diplomsko delo je sestavljeno iz šestih poglavij, in sicer: Uvod, Uporabljena orodja, Komunikacija s tehtnico, Izdelava aplikacije, Predstavitev in uporaba aplikacije kot jo vidi uporabnik, in Sklepne ugotovitve.

Poglavje 2 **Uporabljena orodja**

V tem poglavju so podrobno predstavljena programska orodja in tehnologije, ki smo jih uporabili pri izdelavi aplikacije. Predstavljena pa je tudi strojna oprema, ki smo jo potrebovali za izdelavo in testiranje aplikacije.

2.1 Strojna oprema

Za izdelavo diplomskega dela smo uporabili prenosni računalnik (v nadaljevanju: računalnik) HP Pavilion g6, s sledečimi specifikacijami:

- procesor Intel Core i5-2430M 2.40 GHZ;
- 6 GB DDR3 1333 MHz pomnilnika;
- SATA trdi disk WD 750 GB 5400;
- grafična kartica AMD Radeon HD 6470M 1 GB DDR;
- operacijski sistem Microsoft Windows 8 x 64.

Zahtevana strojna oprema za končne uporabnike aplikacije je enaka zahtevam operacijskega sistema (Windows XP/Vista/7/8). Na primer minimalna zahtevana strojna oprema za uporabo aplikacije na operacijskem sistemu Windows XP:

- procesor: minimalno 1 Ghz, 32-bitni procesor;
- velikost hitrega pomnilnika (RAM): minimalno 256 MB;
- razpoložljiv prostor na disku: minimalno 30 MB;
- minimalna ločljivost zaslona je 800 x 600 pik.

Za razvoj in testiranje aplikacije smo uporabili namizno elektronsko tehtnico Ranger 3000 [4], proizvajalca Ohaus (v nadaljevanju: tehtnica). Tehtnica uporablja enak komunikacijski

protokol kot digitalni tehtalni indikator T71XW [5] cestne tehtnice Ohaus (Slika 1), za katero je bila aplikacija izdelana. Tehtnica (Slika 1) ima sledeče specifikacije:

- nosilnost tehtnice: 30 kg;
- razdelek tehtnice: 0,2 g;
- velikost tehtalne plošče: 150 x 250 mm;
- serijski vmesnik RS-232.



Slika 1: Elektronska tehtnica Ranger 3000 (levo) in digitalni tehtalni indikator T71XW cestne tehtnice Ohaus (desno).

Za vzpostavitev povezave med tehtnico in računalnikom je bilo treba uporabiti USB pretvornik na serijski RS-232 vmesnik (Slika 2) s sledečimi specifikacijami:

- 1 x tip A moški vmesnik;
- 1 x DSUB09 moški vmesnik;
- hitrost prenosa podatkov do 230 kbps;
- LED indikator, dolžina kabla 35 cm in poraba energije 450 mW;
- napajanje iz USB vmesnika.



Slika 2: USB pretvornik na serijski vmesnik RS-232.

2.2 Programska orodja in tehnologije

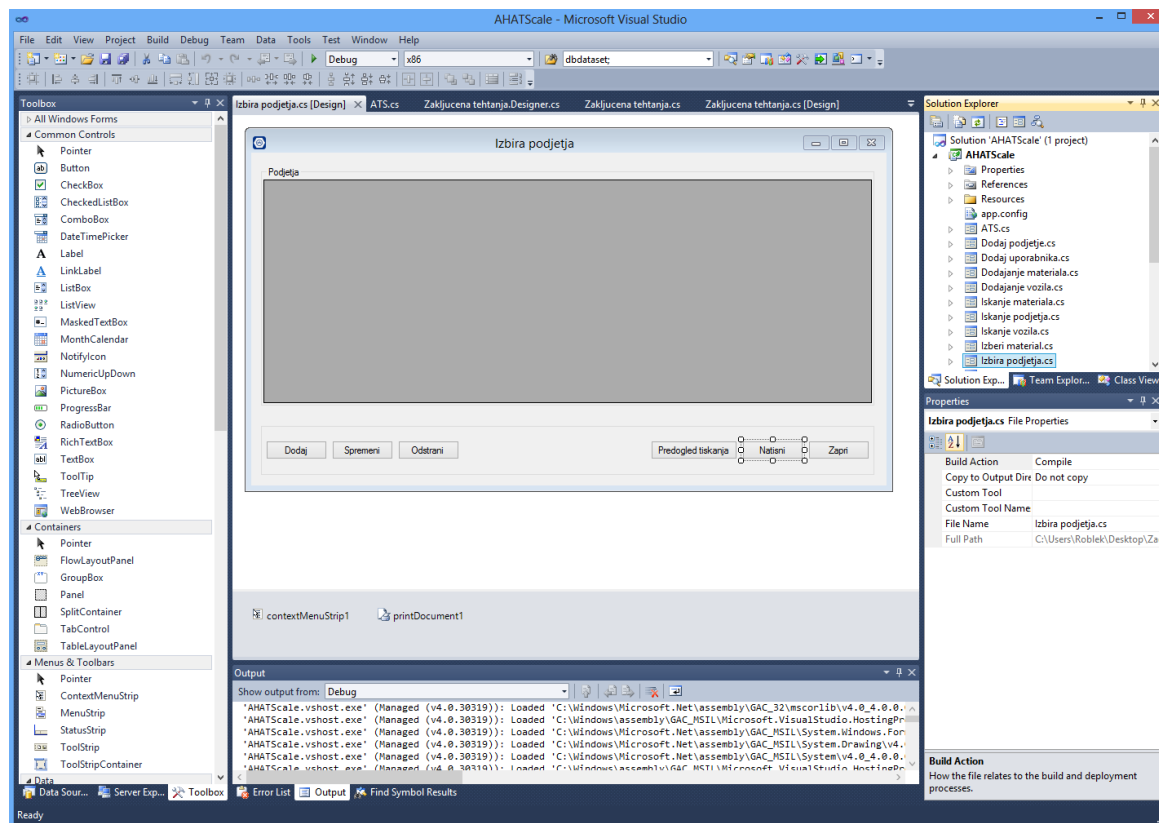
2.2.1 Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 je priljubljeno programsko orodje za razvoj najrazličnejših aplikacij v ogrodju .NET (angl. .NET Framework), ki ga je izdelalo podjetje Microsoft. Microsoft Visual Studio 2010 [6] je integrirano razvojno okolje (angl. Integrated Development Environment, IDE), ki omogoča razvoj konzolnih aplikacij, namiznih aplikacij z grafičnim uporabniškim vmesnikom, spletnih aplikacij, spletnih strani in spletnih storitve. Orodje vsebuje urejevalnik programske kode z zelo učinkovitim dopolnjevanjem kode »IntelliSense«, ki pospeši proces razvoja. Urejevalnik pomaga pri identifikaciji tipkarskih napak, označevanju sintakse, hitrem prikazovanju opisov funkcije itd. Orodje ima vgrajen tudi razhroščevalnik (angl. debugger), s pomočjo katerega lahko hitro odpravimo napake v programski kodi. Prav tako so vgrajene različne komponente za enostavno gradnjo spletnih, namiznih, mobilnih ter tabličnih aplikacij oziroma naprav. Visual Studio podpira programske jezike, kot so: C/C++, VB.NET, C#, F#, Python, Ruby. Prav tako je podprt razvoj CSS, JavaScript, HTML ter XML. Orodje Visual Studio nastopa v več različicah, odvisno od namena uporabe. Najbolj znane različice so Express Edition, Standard Edition in Professional Edition. Trenutno je najnovejša različica Microsoft Visual Studio 2013.

Glavne značilnosti orodja so:

- urejevalnik programske oziroma izvirne kode (angl. source code editor);
- razhroščevalnik (angl. debugger);
- »potegni in spusti« (angl. »drag and drop«), gradnja grafičnih uporabniških vmesnikov;
- razširljivost.

Za izdelavo aplikacije smo uporabili prosto dostopno različico orodja Microsoft Visual Studio 2010 Express (Slika 3), ki ima že vgrajeno zbirko knjižnic .NET Framework 4. Microsoft Visual Studio 2010 Express je na voljo na spletnem naslovu http://www.visualstudio.com/en-us/downloads/download-visual-studio-vs#DownloadFamilies_4.



Slika 3: Programsko orodje Microsoft Visual Studio 2010 Express.

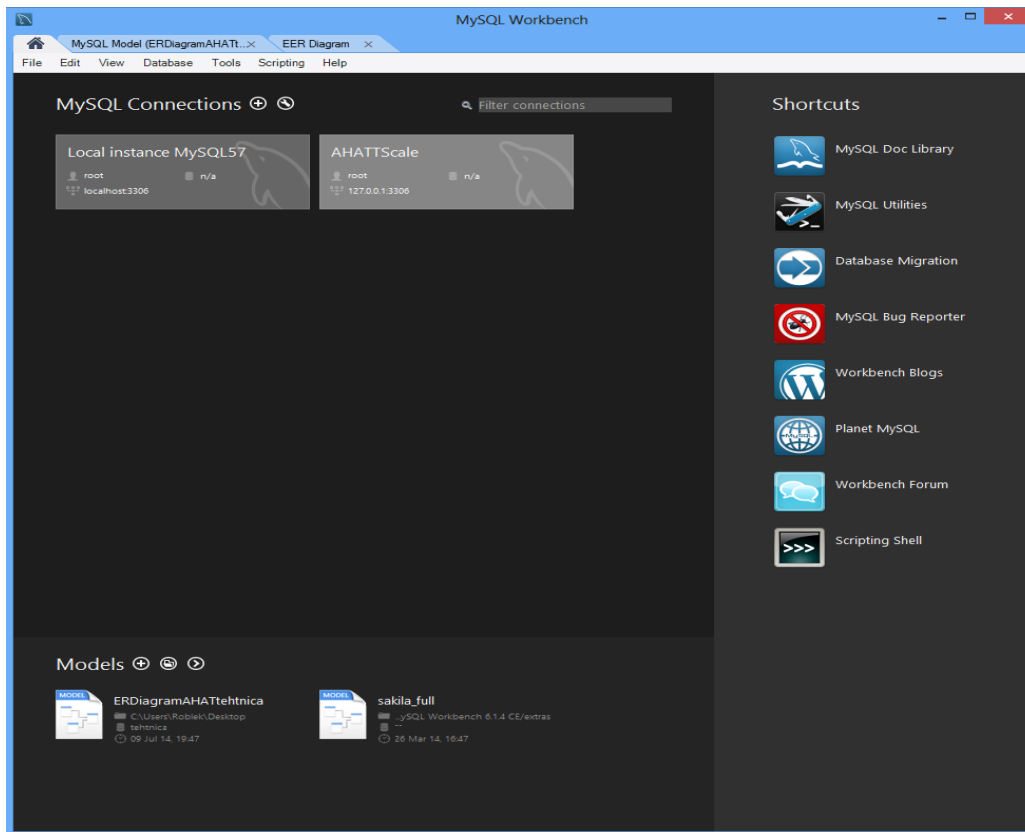
2.2.2 MySQL Workbench

MySQL Workbench [7] je enotno vizualno orodje, namenjeno razvijalcem, načrtovalcem in skrbnikom podatkovnih baz. Izdelalo ga je podjetje Sun Microsystems. MySQL Workbench omogoča modeliranje podatkov, SQL razvoj in je celovito administrativno orodje za konfiguracijo strežnikov, upravljanje uporabnikov, »backup« in še veliko več [8]. Orodje je na voljo za Windows, Linux in Mac operacijski sistem. Obstajata dve različici MySQL Workbench orodja, prosto dostopna in plačljiva različica.

Glavne funkcionalnosti orodja MySQL Workbench:

- vizualna predstavitev tabel, pogledov, shranjenih procedur/funkcij in tujih ključev;
- žive zbirke podatkov in generiranje modela iz skript SQL;
- sinhronizacija modela in podatkovnega skladišča;
- generiranje skript SQL;
- izvoz skript SQL CREATE.

Za izdelavo in implementacijo podatkovnega modela smo uporabili prosto dostopno različico orodja MySQL Workbench 6.1.4 (Slika 4). MySQL Workbench 6.1.4 je na voljo na spletnem naslovu <http://dev.mysql.com/downloads/workbench/>.



Slika 4: Programsko orodje MySQL Workbench 6.1.4.

2.2.3 MySQL

MySQL je priljubljen odprtokodni sistem za upravljanje s podatkovnimi bazami [9]. Je klasična (relacijska) zbirka podatkov, ki lahko deluje kot samostojen podatkovni strežnik ali pa v kombinaciji z drugimi tehnologijami [10]. Poizvedbe v MySQL podatkovni bazi se vršijo z jezikom SQL, ki je najbolj razširjen in uporabljen jezik za pisanje poizvedb. Določen je s standardom ANSI/ISO SQL.

Pred približno devetnajstimi leti je MySQL v jezikih C in C++ razvilo švedsko podjetje MySQL AB in ga ponudilo trgu pod pogoji uporabe javne licence GNU (<http://www.gnu.org>). Postal je svetovno najbolj priljubljena baza odprtega tipa, katere uporaba po zaslugi visoke zanesljivosti, hitrega delovanja, prilagojenosti in nenehnega razvoja še vedno strmo raste [10].

Strežnik MySQL lahko upravljate na dva načina: enostavno, z grafičnim uporabniškim vmesnikom (MySQL Workbench, phpMyAdmin, MySQL Front itd.), ali pa z ukazno vrstico, kar je počasneje in pri velikih zbirkah podatkov precej nepregledno. Za boljše razumevanje MySQL-a je priporočljivo poznavanje upravljanja v obeh načinih [10].

Pomembne lastnosti MySQL [11]:

- napisan je v jezikih C in C++;
- testiran z veliko vrstami različnih orodij;
- dela na več različnih operacijskih sistemih (Linux, Windows, OS/X ...);
- za prenosljivost med operacijskimi sistemi uporablja GNU Automake, Autoconf in Libtool;
- API-ji (aplikacijski programske vmesniki) so na voljo za jezike C, C++, Eiffel, Java, Perl, PHP, Python, Ruby in TCL;
- lahko uporablja večprocesorske sisteme;
- uporablja zelo hitre tabele Btree diska (MyISAM), s kompresijo indeksa;
- relativno lahko se doda še eno hranilno napravo;
- zelo hiter omrežni sistem za razporejanje pomnilnika;
- zelo hitri spajalni členi, ki uporabljajo optimiziran onesweep multijoin;
- tabele Inmemory hash se lahko uporabljajo kot začasne tabele;
- funkcije SQL so izboljšane z uporabo visoko optimizirane razredne knjižnice in naj bi bile tako hitre, kot je le mogoče;
- koda MySQL je testirana z orodji Purify (komercialni detektor uhajanja spomina), Valgrind (odprtokodno orodje za izgradnjo dinamičnih orodij za analizo);
- strežnik je na voljo kot ločen program za uporabo v omrežnem okolju strežnik – odjemalec.

2.2.4 Programski jezik C#

C# (C sharp) je programski jezik [12] s poudarkom na objektno orientiranem pristopu, tipih in generičnih tipih, kar omogoča boljšo strukturiranost kode ter lažje programiranje. Izdelalo ga je podjetje Microsoft leta 2000. Skozi leta je podjetje jezik izboljševalo, zadnja različica jezika je prišla leta 2013. Jezik je nekakšna kombinacija jezika C in C++. Čeprav C# temelji na C++, pa vsebuje številne lastnosti, značilne za Javo [19]. Uporaba programskega jezika C# se skozi čas povečuje, kar je posledica enostavne uporabe ter možnost uporabe knjižnic iz ogrodja .NET Framework. Velika prednost opisanega programskega jezika je, da je podprt v vseh verzijah orodja Microsoft Visual Studio.

Glavne značilnosti jezika C#:

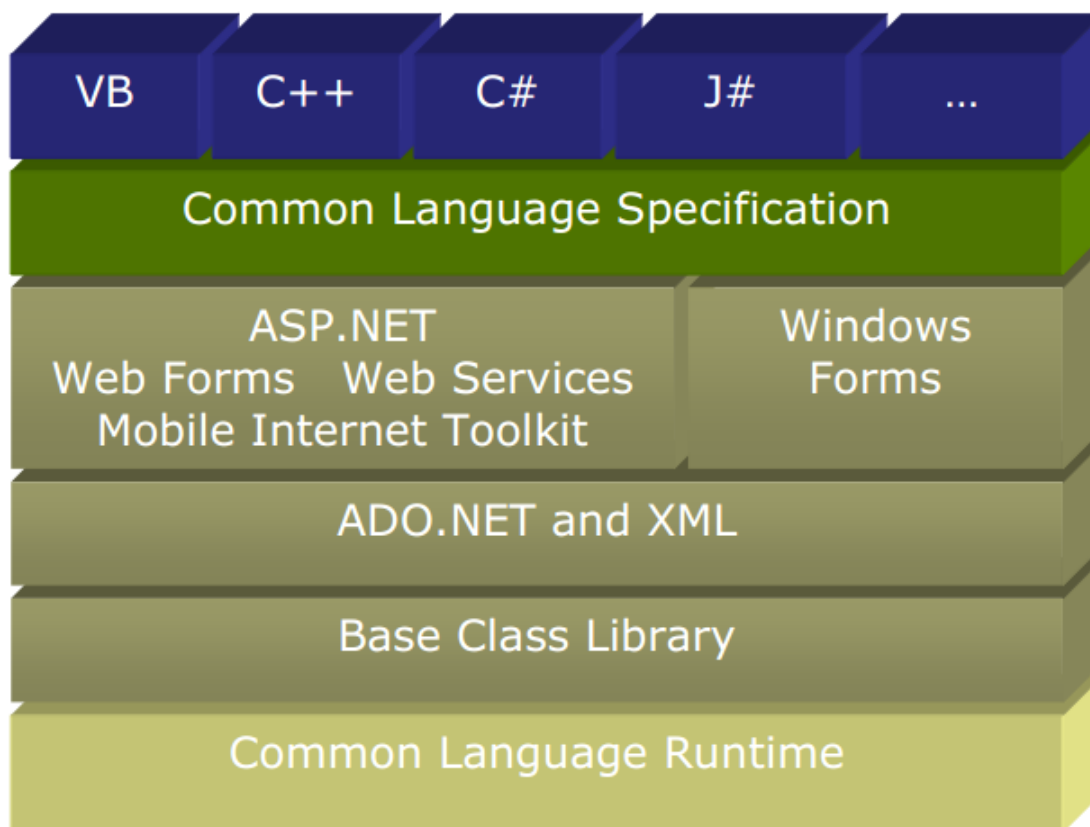
- preprost,
- varen,
- moderen,
- objektno orientiran,
- visoko kvaliteten,
- internetno usmerjen.

2.2.5 .NET Framework

Ogrodje .NET Framework [13, 14] je Microsoftov programiran model upravljanja programske kode (angl. managed code) in računalniško okolje (angl. platform) za razvoj aplikacij na Microsoftovih odjemalcih, strežnikih in mobilnih ter vgradnih napravah. Osnovna arhitektura, ki je prikazana na Sliki 5, se kljub novim različicam ne spreminja. Orodje ponuja pregledno konsistentno objektno orientirano programiranje s poudarkom na varnosti ter visoki produktivnosti. Vzporedno z različicami ogrodja .NET Framework so tudi jeziki, s katerimi lahko programiramo v ogrodju .NET Framework (C#, J#, C/C++ in Visual Basic), doživljali ustrezne dopolnitve. Skladno z različicami ogrodja .NET Framework se je razvijalo tudi programsko okolje Microsoft Visual Studio.

Ogrodje .NET Framework [13] obsega skupno izvajalno okolje (angl. Common Language Runtime, CLR) in zbirko raznovrstnih razrednih knjižnic .NET Framework class library, ki vsebuje zbirke programskih komponent za dostop do podatkov in podatkovnih servisov (angl. ActiveX Data Object for NET, ADO.NET), okolje za razvoj interaktivnih spletnih aplikacij, ki

temeljijo na strežniških kodah (angl. Active Server Pages for NET, ASP.NET), okna Windows (angl. Windows forms) in grafične podsisteme za uporabljanje uporabniških vmesnikov, ki temeljijo na operacijskem sistemu Windows (angl. Windows Presentation Foundation, WPF). Razredi, ki jih knjižnice vsebujejo, so hierarhično organizirani v tako imenovane imenske prostore (angl. name-spaces). Skupno izvajalno okolje nudi storitev za izvajanje programov. Za izdelavo aplikacije smo uporabili ogrodje Microsoft .NET Framework 4.0.



Slika 5: Zgradba .NET Framework-a.

Poglavje 3 **Komunikacija s tehtnico**

V tem poglavju bomo opisali serijski vmesnik RS-232, preko katerega je tehtnica povezana z računalnikom, kot tudi protokol, s katerim se lahko računalnik preko serijskega vmesnika sporazumeva s tehtnico.

Na področju računalništva je serijski vmesnik RS-232 fizični vmesnik za zaporedno oziroma serijsko komunikacijo, skozi katero se podatki pretakajo ven ali noter, en bit naenkrat. Skoraj čez vso zgodovino osebnih računalnikov so se podatki pretakali skozi zaporedna oziroma serijska vrata v naprave, kot so modemi, terminali in različne periferne naprave (tipkovnice, miške, videoterminali, tiskalniki in podobno).

Ob prodoru novejših komunikacijskih tehnologij delež serijskih komunikacijskih povezav v svetu povsod močno upada, razen v industrijskih okoljih, kjer naprave (tehtnice, skenerje, krmilnike) še vedno pogosto povezujemo s serijskimi vmesniki RS-232 [15].

3.1 Standard RS-232

Standard [17] so razvili leta 1969 v ZDA pod okriljem mednarodne organizacije EIA (angl. Electronics Industries Association) za potrebe komuniciranja med DTE (angl. Data Terminal Equipment) in DCE (angl. Data Circuit-Terminating Equipment) pri nizki hitrosti zaporednega prenosa. Standard določa mehanske lastnosti priključnih konektorjev, električnih lastnosti vhodov in izhodov, njihove funkcije ter časovne poteke pripadajočih signalov [16]. Standard RS-232 je najpreprostejši in najbolj znan standard, ki omogoča dvosmeren prenos podatkov po liniji. Novejša verzija tega standarda je RS-232C standard.

V današnjem času je komunikacijo po standardu RS-232 zamenjal prenos podatkov po standardu USB. Kljub temu serijski vmesnik RS-232 še vedno najdemo v napravah, ki zahtevajo robustno povezavo na krajših razdaljah, komunikacijo od točke do točke, kot so na primer senzorji, industrijske naprave, mrežna stikala, modemi, baterije za neprekinjeno napajanje, skenerji, tehtnice in podobno.

3.1.1 Napetosti logičnih nivojev

Logični nivoji so pri RS-232 predstavljeni s pozitivnimi in negativnimi napetostmi (Tabela 1). Negativna napetost predstavlja logično »1« in pomeni neaktivno stanje (»OFF«), pozitivna napetost pa logično »0« in pomeni aktivno stanje (»ON«). Zaradi možnih daljših (serijskih) povezovalnih linij so minimalne napetosti, definirane pri sprejemniku, nižje od tistih, definiranih pri oddajniku. Pri sprejemniku je vhodna napetost, višja od +3 V, sprejeta kot logična »0«, nižja od -3 V pa kot logična »1«. Pri oddajniku je minimalna napetost +5 V in -5 V. Maksimalna dovoljena napetost je +/- 15 V (pri računalniku je ta napetost +/- 12 V) [18].

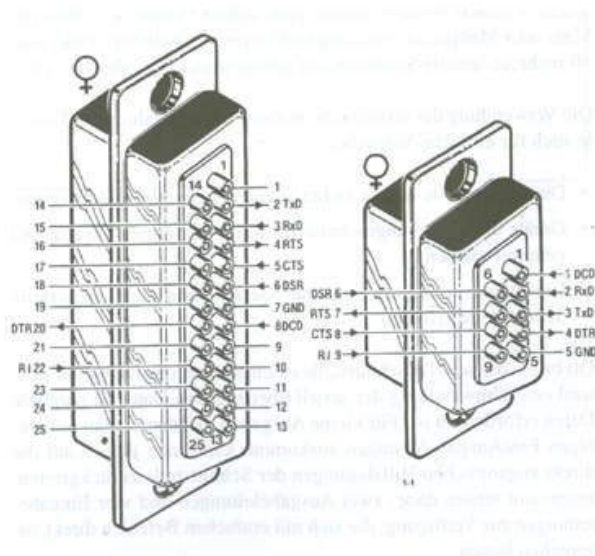
| Podatkovna linija | | Kontrolna linija | |
|-------------------|-------------------|------------------|------------|
| Logična 1 | Logična 0 | OFF | ON |
| Znak (mark) | Presledek (space) | Nizek nivo | Visok nivo |
| Napetost: | | | |
| -3 V -15 V | -3 V -15 V | -3 V -15 V | -3 V -15 V |

Tabela 1: Napetostni nivoji za podatkovne in kontrolne signale.

3.1.2 Signali RS-232

Standard RS-232 definira 25 signalov, vendar računalnik in druge naprave večinoma uporabljajo največ 9 signalov, to so tisti signali, ki so najpomembnejši.

Standard RS-232 predpisuje tudi obliko konektorja in funkcijo vsakega priključka na njem. Konektor obstaja v dveh izvedbah (Slika 6): 25-polni moški/ženski konektor tipa D (krajše DB25), ki ima 25 priključkov, in 9-polni moški/ženski konektor tipa D (krajše DB9), ki ima 9 priključkov. V Tabeli 2 sta opisana oba.



Slika 6: RS-232 konektor DB25 in DB9 s priključki.

| D25 | DB9 | Funkcija | Oznaka | Smer DTE | Smer DCE |
|-----|-----|-----------------------|------------------------------------|----------|----------|
| 1 | - | Oklop | | | |
| 2 | 3 | Oddajni podatki | Transmitted Data (TD) | Izhod | Vhod |
| 3 | 2 | Sprejemni podatki | Received Data (RD) | Vhod | Izhod |
| 4 | 7 | Zahteva za oddajo | Request To Send (RTS) | Izhod | Vhod |
| 5 | 8 | Pripravljen za oddajo | Clear To Send (CTS) | Vhod | Izhod |
| 6 | 6 | Podatki pripravljeni | Data Set Ready (DSR) | Vhod | Izhod |
| 7 | 5 | Masa | Masa (GND) | | |
| 8 | 1 | Nosilec zaznan | Carrier Detect (CD) | Vhod | Izhod |
| 20 | 4 | Terminal pripravljen | Data Terminal Ready (DTR) | Izhod | Vhod |
| 22 | 9 | Klicni signal | Ring Indicator (RI) | Vhod | Izhod |

Tabela 2: Priključki in smeri signalov RS-232.

V Tabeli 2 so navedeni samo najpomembnejši signali, to so tisti, ki jih ima 9-polni konektor (DB9).

Pri dvosmerni RS-232 komunikaciji potrebujemo vsaj tri signale:

- TD (označuje se tudi TX ali TXD), prenaša podatke od naprave tipa DTE k napravi tipa DCE;
- RD (tudi RX ali RXD), prenaša podatke od naprave tipa DCE k napravi tipa DTE;
- GND (tudi SG ali SGND), masa signala.

Ostali signali so namenjeni za kontrolo prenosa in običajno sporočajo pripravljenost naprave za prenos podatkov. Ti signali nastopajo v parih: RTS/CTS in DTR/DSR, njihovo uporabo pa določa standard.

Signala RTS/CTS sporočata pripravljenost naprave za prenos podatkov. RTS pove, da naprava tipa DTE (v nadaljevanju: DTE) želi pošiljati podatke k napravi tipa DCE (v nadaljevanju: DCE). DTE mora biti vedno pripravljen na sprejem, saj v obratni smeri ni ustreznega signala. Med normalnim delovanjem bo RTS na visokem logičnem nivoju. Ko ima DTE podatke pripravljene za prenos, postavi RTS na nizek logični nivo in čaka odgovor CTS linije od DCE. Po začetku prenosa DTE postavi RTS zopet na visok nivo. Pri dvosmernem načinu prenosa (angl. full duplex) ali enosmernem načinu prenosa (angl. simplex) tega signala ne rabimo. CTS signalizira, da je DCE pripravljen na sprejem podatkov od DTE. Pri normalnem delu je tudi CTS na visokem logičnem nivoju. Ko DCE postavi CTS na nizek nivo, lahko DTE začne s pošiljanjem podatkov [18].

Signala DTR/DSR se uporabljata za zagotavljanje informacij o stanju linije, priključene na DCE. DTE postavi DTR, ko želi, da se DCE poveže na komunikacijski kanal. DCE postavi DSR signal na visok nivo, ko je priključen. DSR je na nizkem nivoju v primeru, ko DCE ni priključen na linijo ali pa se ta prekine. DTE lahko postavi DTR kot odziv na RI (angl. Ring Indikator) signal. Na ta način pove DCE, naj odgovori na prihajajoč poziv. V nekaterih vezjih sta signala DTR in DSR stalno na visokem nivoju in tako sporočata, da so naprave prisotne in pod napetostjo [18].

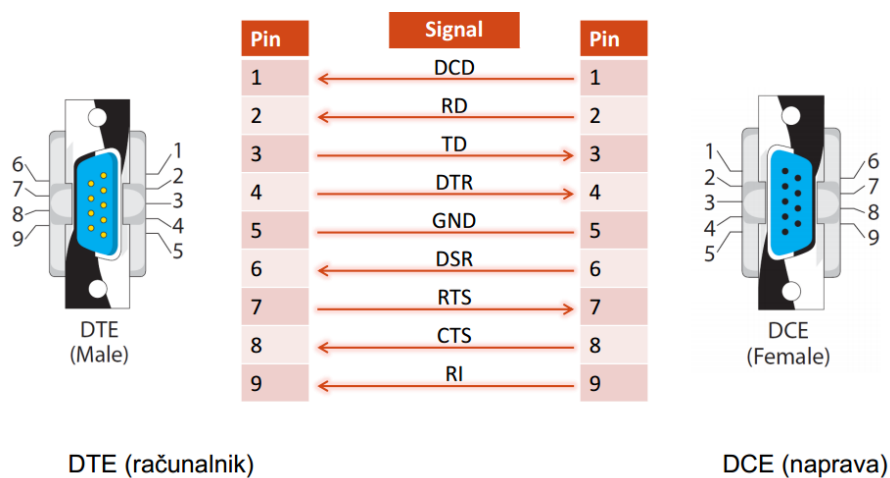
3.1.3 Null modem

Ideja, imeti dva različna vmesnika za DTE in DCE napravo, je bila verjetno razumska okrog leta 1960. Danes svet ni več tako »enostaven«, saj bi uporabniki s serijskim vmesnikom radi počeli vse mogoče. Da so stvari še bolj zapletene, je IBM, ko je v prvi računalnik vključil

serijski vmesnik, le-tega definiral kot DTE, čeprav mu danes pripada jasna vloga DCE. V tistih časih so imeli računalnik za napravo, ki se bo kot terminal povezoval na večje »mainframe« računalnike. Na računalnik tako lahko povežemo modem, saj je to naprava tipa DCE, ne moremo pa povezati dveh računalnikov med seboj. Če uporabimo običajen kabel, imata napravi skupaj povezani TD in RD liniji in ne vidita druga druge. Rešitev je v uporabi »null modem« kabla, ki simulira povezavo DTE/DCE tako, da zamenja komplementarne in kontrolne signale.

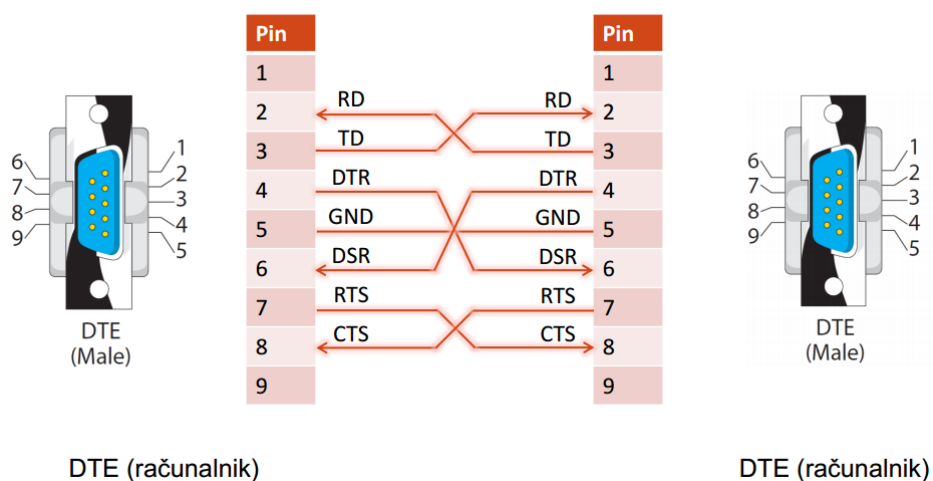
Obstaja več načinov povezav za 9-polni konektor (DB9):

- povezava (RS-232) med DCE in DTE (Slika 7);



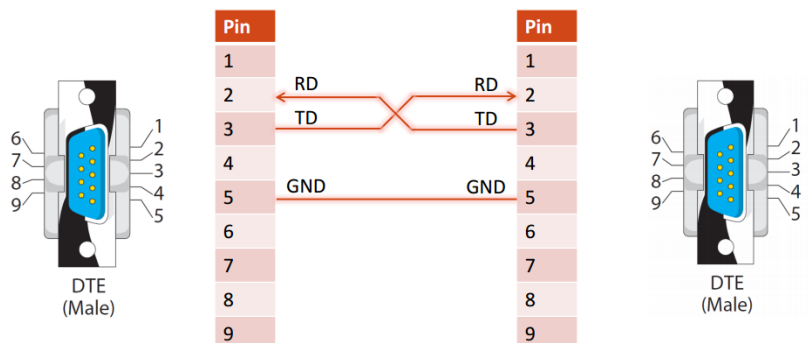
Slika 7: Serijska povezava med DCE in DTE.

- povezava (RS-232) med DTE in DTE (Slika 8);



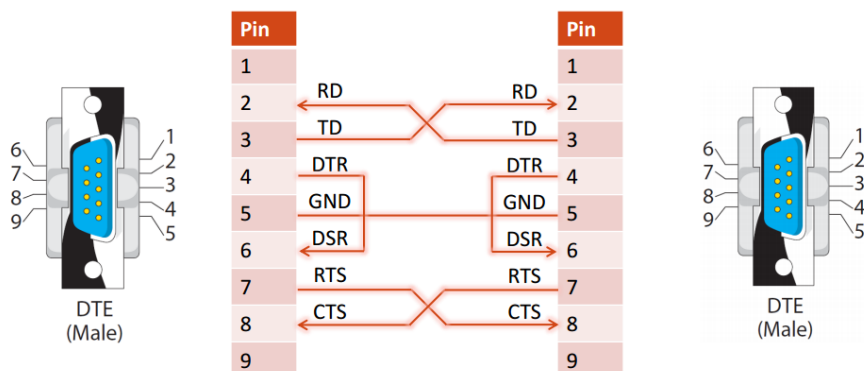
Slika 8: Serijska povezava med DTE in DTE (null modem).

- povezava (RS-232) med DTE in DTE brez usklajevanja (Slika 9);



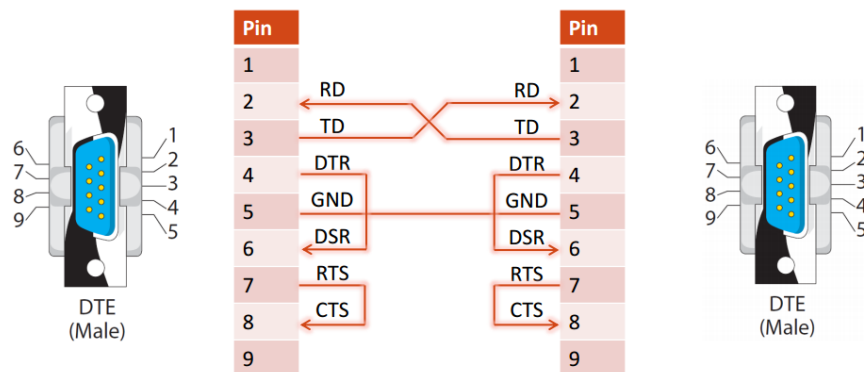
Slika 9: Serijska povezava med DTE in DTE brez usklajevanja (null modem).

- povezava (RS-232) med DTE in DTE z delnim usklajevanjem (Slika 10);



Slika 10: Serijska povezava med DTE in DTE z delnim usklajevanjem (null modem).

- povezava (RS-232) med DTE in DTE s samo-usklajevanjem (Slika 11).



Slika 11: Serijska povezava med DTE in DTE s samo-usklajevanjem (null modem).

3.1.4 Okvir RS-232

Po standardu RS-232 se informacije pošiljajo v obliki nizov bitov na fizično povezavno linijo, podatki bitov pa sestavljajo nekakšno digitalno besedo. RS-232 standard omogoča spremenljivo dolžino besede, in sicer od 5 do 8 bitov, kar pomeni število informacij v paketu. Zelo pomembno je, da se oddajnik in sprejemnik identično nastavita, ker drugače prihaja do napak zaradi časovne neusklajenosti. Za popoln prenos pa standard določa bite za sinhronizacijo in ugotavljanje napak, vendar zaradi tega pride do časovnih izgub.

Hitrost prenosa izražamo kot število podatkovnih bitov, prenesenih v eni sekundi in jo imenujemo bitna hitrost (angl. bit rate) v enotah bps (angl. bitov/s, bit per second). Baudna stopnja (angl. baud rate) se nanaša na število signalnih ali simbolnih sprememb prenašalnega signala v eni sekundi na prenosni liniji. Pri sistemu, ki uporablja le dva nivoja (binarni signal), je bitna stopnja enaka baudni. Hitrost prenosa vrat in hitrost prenosa naprave se morata ujemati. Sposobnost, da se določi hitrost prenosa, še ne pomeni, da bo posledično povezava delovala. Vse bitne hitrosti niso mogoče pri vseh serijskih vratih. Splošno podprte bitne hitrosti so: 75, 110, 300, 1200, 2400, 4800, 9600, 19.200, 38.400, 57.600 in 115.200 bitov/s.

Start in stop bita sta nepogrešljiv člen pri asinhronem prenosu podatkov, saj je čas med pošiljanjem posameznih znakov različen. Start bit sprejemniku da vedeti, kdaj naj začne brati oziroma označuje začetek novega znaka. Start bit ima logični nivo »0«, kar se razlikuje od logičnega nivoja »1«, ki je ob mirovnem stanju na liniji, ter se z njim vedno prične RS-232 paket. Stop bit pa zapira okvir sporočila in je negativnega napetostnega nivoja oziroma enak prosti liniji. Večina sistemov uporablja en stop bit, nekateri pa tudi dva.

Podatkovnih bitov na znak je lahko 5 (za Baudotovo kodo), 6 (redko uporabljena), 7 (za pravi ASCII), 8 (za veliko večino podatkov, ker je velikost enaka velikosti bajta) ali pa 9 (redko uporabljena). 8 podatkovnih bitov je skoraj univerzalno uporabljenih v novejših aplikacijah, 5 ali 7 bitov pa po navadi le pri starejši opremi, kot so teleprinterji.

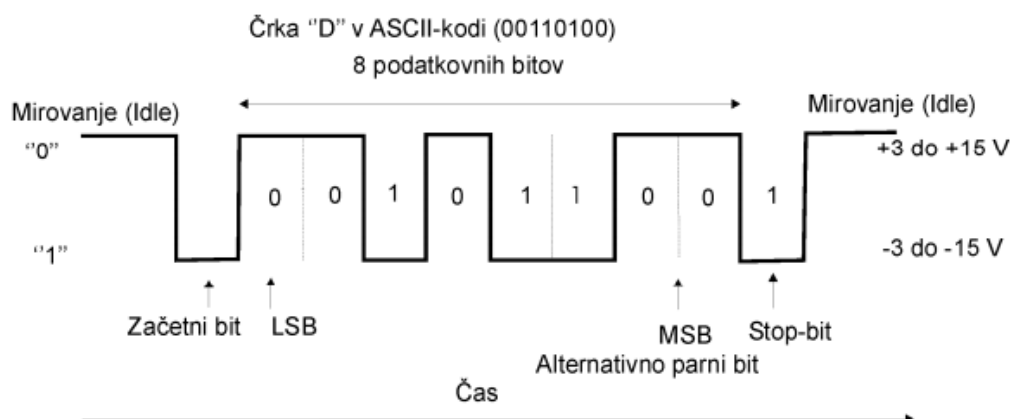
Večina serijskih komunikacijskih modelov pošilja podatkovne bite znotraj enega bajta, s standardom LSB naprej. Temu standardu se reče tudi »little endian«. Prav tako mogoč, a redko uporabljen, je »big endian« ali MSB naprej. Vrstnega reda bitov po navadi ni mogoče nastaviti znotraj vmesnika zaporednih vrat. Za komunikacijo s sistemi, ki zahtevajo drugačen vrstni red bitov, kot je lokalno privzeto, lahko lokalna programska oprema spremeni vrstni red bitov znotraj vsakega bajta, tik pred pošiljanjem in tik pred prejemanjem.

Pariteta je enostavna in na žalost tudi slaba zaščita pred napakami pri prenosu podatkov, saj opazi samo eno napako.

Ločimo pet načinov ugotavljanja paritete:

- brez paritete (angl. none) – (»N«);
- liha pariteta (angl. odd) – (»O«);
- soda pariteta (angl. even) – (»E«);
- oznaka (angl. mark) – (»M«);
- prostor (angl. space) – (»S«).

Paritetni bit v vsakem znaku se lahko nastavi na nič (»N«), liho (»O«), sodo (»E«), oznako (»M«) ali prostor (»S«). N pomeni, da se ne pošlje noben paritetni bit, »M« pomeni, da je paritetni bit vedno nastavljen na pogoj signala oznake (logična »1«), in prav tako »S« pomeni, da je paritetni bit nastavljen na pogoj signala prostora. Razen redkih aplikacij, ki uporabljajo deveti paritetni bit za neko obliko naslavljanja ali posebne signalizacije, sta »M« in »S« pariteti zelo redki, saj ne dodata nobenih podatkov o napaki. »O« pariteta je bolj uporabna kot »E«, ker zagotavlja, da se zgodi vsaj ena sprememba stanja v vsakem znaku, kar naredi »O« pariteto bolj zanesljivo. Najbolj pogosta nastavitev paritete pa je »N«, pri kateri zaznavanje napak obvladuje komunikacijski protokol. Paritetni bit se določa na osnovi seštevek enic v znaku. Seštevek enic mora biti pri »E« pariteti sodo, pri »O« pariteti pa liho število. V nasprotnem primeru računalnik doda enice, da izpolni pogoj. Sprejemnik sešteje enice in preveri paritetni bit. Če se ne ujemata, sporoči oddajniku, naj ponovi prenos tega znaka. Večina sistemov, ki uporabljajo 7 bitov na znak, uporablja »E« pariteto. Pri 8-bitni predstavitvi znaka se paritetni bit ne uporablja. Kot primer (Slika 12) vzemimo veliko črko D, ki ima ASCII znak 1000100P. Seštevek enic je sodo število, zato pri »O« pariteti dodamo enico P = 1, pri sodi pa je P = 0.



Slika 12: Primer tipičnega okvirja RS-232.

3.1.5 Napake pri prenosu

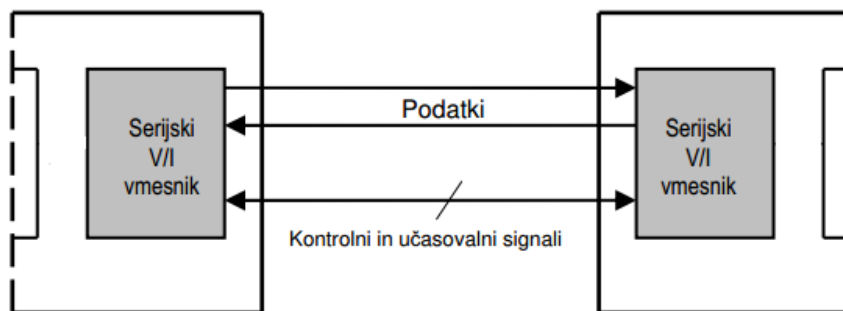
Omeniti moramo še napake, ki jih lahko zazna vezje, ki skrbi za prenos podatkov. Te napake so lahko:

- **napake okvirja** (angl. framing error): to je napaka, ki jo zazna sprejemno vezje, če ugotovi, da start in stop bit nista na svojih mestih;
- **prekinitvena napaka** (angl. break error): sprejemnik je zaznal, da je sprejemna linija »nenormalno dolga« (vsaj toliko časa, kolikor se potrebuje za prenos 8 bitov) na nizkem logičnem nivoju;
- **napaka prekoračitve sprejemnega pomnilnika** (angl. overrun error): do te napake pride pri prepočasnem sprejemanju podatkov iz začasnega pomnilnika sprejemnika. Napaka pove, da so zadnje prispeli podatki po vsej verjetnosti prepisali tiste, ki so prišli pred njim;
- **napaka paritete**: vzrok te napake so lahko različne nastavitve hitrosti ali ugotavljanja paritete pri sprejemniku ali oddajniku ali napake, ki so nastale pri prenosu podatkov (na primer: zaradi šuma).

3.2 Komunikacija med tehtnico in računalnikom

Da računalnik lahko prebere težo iz tehtnice, mora biti tehtnica povezana z računalnikom. Tako tehtnica kot tudi računalnik morata imeti vgrajen vmesnik za serijsko komunikacijo (RS-232), ki skrbi za pravilno delovanje in zanesljiv prenos podatkov od ene naprave k drugi in obratno.

Računalnik komunicira s tehtnico preko tako imenovanih serijskih vhodno/izhodnih vrat (angl. com port). Če računalnik nima vgrajenih serijskih vhodno/izhodnih vrat, uporabimo pretvornik iz USB v RS-232. Pretvornik USB v RS-232 omogoča povezavo naprav s serijskim vmesnikom RS-232 preko USB vhoda z računalnikom. Serijski vmesnik na računalniku in tehtnici je običajno tipa DTE. Tip naprave pove, kateri signali so vhodni in kateri izhodni na priključku naprave. Vsa imena signalov se nanašajo na perspektivo DTE. Tako je na primer TD izhodni signal DTE in vhodni DCE. Sprejemna linija RD je vhodni na DTE in izhodni na DCE [18]. Komunikacija prenosa med tehtnico in računalnikom je tipa poln dvosmerni (angl. full-duplex). Pri tej vrsti komunikacije se podatki oddajajo in sprejemajo hkrati. Primer polne dvosmerne komunikacije med računalnikom in tehtnico prikazuje Slika 13.



Slika 13: Primer serijske povezave med računalnikom in tehtnico.

3.2.1 Serijski vmesnik tehtnice Ohaus

Serijski vmesnik tehtnice [20] omogoča dvosmerno izmenjavo podatkov med tehtnico in računalnikom. Oblika prenosa signala je asinhrona. Za sporazumevanje tehtnice in računalnika je potrebna sinhronizacija, ki je določena z začetnim (start) in zaključnim (stop) bitom. Vmes med bitoma si sledi 8 podatkovnih bitov, od bita z najmanjšo težo (LSB) do bita z največjo težo (MSB). Podatkovni okvir (Slika 14) je podrobneje opisan in prikazan v razdelku 3.1.4.



Slika 14: Oblika prenosa podatka pri komunikaciji med tehtnico in računalnikom.

3.2.1.1 Tehnični podatki vmesnika

V Tabeli 3 so navedeni tehnični podatki serijskega vmesnika tehtnice.

| | | |
|---|--------------------------|---|
| 1 | Prenosno omrežje | Serijsko/Asinhrono |
| 2 | Hitrost prenosa podatkov | 300, 600, 1200, 2400, 4800, 9600, 19.200 bps |
| 3 | Koda prenosa | ASCII kode (8/7 bitov) |
| 4 | Signal | Visoka raven (podatki logična »0«) +5 do +15 V Nizka raven (podatki logična »1«) -5 do -15 V |
| 5 | Bitne nastavitve | Začetni bit: 1 bit Podatkovni bit: 8/7 bitov Paritetni bit: 0/1 bit Končni biti: 2/1 bit |
| 6 | Pariteta | Soda/liha/brez parnosti |

Tabela 3: Tehnični podatki serijskega vmesnika tehtnice.

3.2.1.2 Priključki in uporabljeni signali

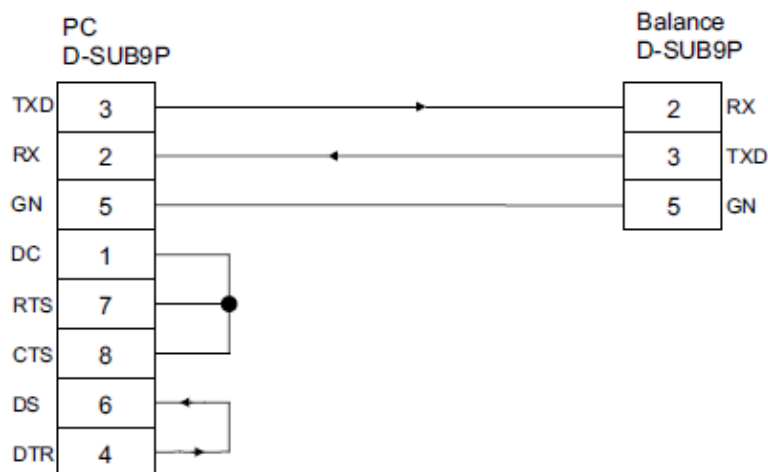
Dodelitev priključkov na konektorju tehtnice prikazuje Tabela 4. Čeprav imamo DB9 konektor za komunikacijo med tehtnico in računalnikom, rabimo le 3 priključke. To so priključki 2, 3 in 5, druge priključke pa je priporočljivo povezati skupaj.

| Številka priključka | Signal | Vhod/izhod | Funkcija |
|---------------------|--------|------------|---------------------|
| 2 | RD | Vhod | Prejemanje podatkov |
| 3 | TD | Izhod | Oddajanje podatkov |
| 5 | GND | | |

Tabela 4: Razpored priključkov na konektorju.

3.2.1.3 Kabel vmesnika

Za povezavo med tehtnico in računalnikom izdelamo povezovalni kabel (serijski kabel), ki omogoča komunikacijo med omenjenima napravama z uporabo standarda za serijski vmesnik RS-232. Slika 15 prikazuje shemo povezovalnega kabla.



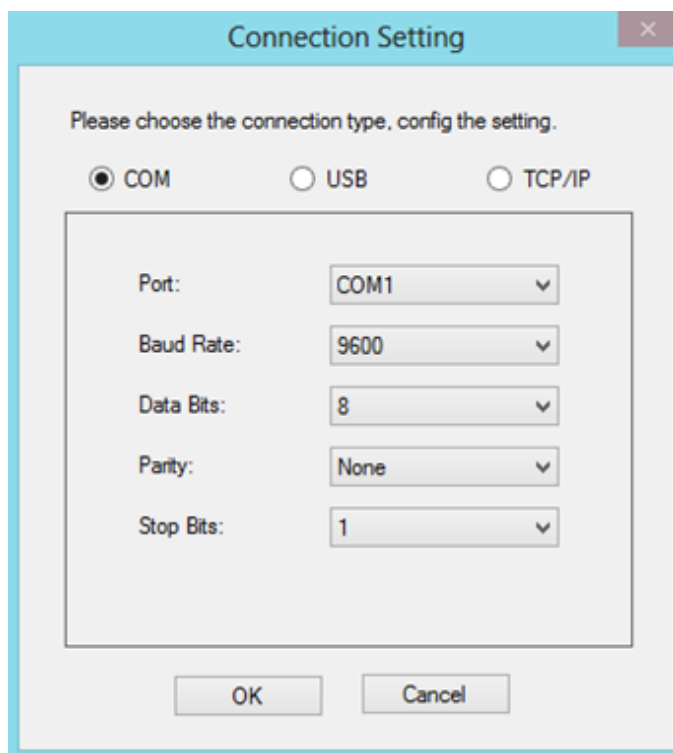
Slika 15: Shema povezovalnega kabla.

3.2.1.4 Parametri komuniciranja

Standard RS-232 določa parametre komuniciranja. Vrednosti osnovnih komunikacijskih parametrov na tehtnici lahko nastavljamo, in sicer:

- hitrost prenosa – takt (hitrost takta, tipične vrednosti 300, 600, 1200, 2400, 4800, **9600** in 19.200 bps);
- podatkovni bit – število podatkovnih bitov (7 in **8**);
- pariteta – prisotnost bita parnosti in vrsta parnosti (**brez parnosti**, soda in liha);
- stop bit – število zaključnih bitov (**1** in 2);
- nadzor pretoka – krmiljenje pretoka podatkov (brez, Xvklop/Xizklop, strojna oprema).

Pred prenašanjem podatkov morata imeti tehtnica in računalnik te vrednosti enako nastavljene. Slika 16 prikazuje nastavitve parametrov serijskega vmesnika tehtnice oziroma njihove vrednosti.



Slika 16: Nastavitve parametrov serijske povezave.

3.2.1.5 Protokol tehtnice

S protokolom definiramo, na kakšen način naj aplikacija sprejema podatke in iz njih izlušči ustrezne podatke o masi.

Podatki, poslani iz tehtnice, so zapisani v standardnem ASCII formatu. Nizi podatkov o tehtanju, ki jih tehtnica pošilja na računalnik, so zaključeni z znakom za konec vrstice (CR) in z znakom za začetek nove vrstice (LF). Obliko podatka, ki ga pošlje tehtnica, prikazuje Tabela 5.

| Polje | Dolžina |
|-------------------------|-----------|
| Oznaka (angl. Label) | 11 znakov |
| Presledek (angl. Space) | 1 znak |
| Teža (angl. Weight) | 9 znakov |
| Presledek (angl. Space) | 1 znak |
| Enota (angl. Unit) | 5 znakov |

| | |
|--|---------|
| Presledek (angl. Space) | 1 znak |
| Stabilnost (angl. Stability) | 1 znak |
| Presledek (angl. Space) | 1 znak |
| Neto/bruto (angl. G/N) | 3 znaki |
| Presledek (angl. Space) | 1 znak |
| Zaključni znaki (angl. Termination Characters) | 4 znaki |

Tabela 5: Oblika podatka, ki ga pošlje tehtnica.

Podatkovni protokol tehtnice je določen z:

- dolžino niza,
- terminatorjem niza,
- začetno pozicijo mase,
- dolžino mase,
- tipom mase,
- začetno pozicijo tipa mase,
- dolžino tipa mase,
- nizom za neto maso,
- nizom za bruto maso,
- enoto,
- začetno pozicijo enote,
- dolžino enote,
- začetno pozicijo tare,
- dolžino tare,

- začetno pozicijo stabilnosti,
- dolžino stabilnosti,
- nizom za stabilno maso,
- nizom za nestabilno maso.

Tehtnica se odziva tudi na različne zunanje ukaze. Tako je na primer mogoče iz računalnika tehtnico tarirati (ASCII 84) in ničlirati (ASCII 90). V Tabeli 6 so prikazane in opisane kontrolne kode in znaki, ki jih tehtnica pošilja na računalnik ali sprejema od računalnika. Vsakemu poslanemu ukazu (»koda« ali »znak«), preko katerega nam tehtnica vrne podatek o masi, moramo dodati kontrolni znak za konec vrstice (CR) in kontrolni znak za začetek nove vrstice (LF).

| Tip podatka | Vsebina | Funkcija | ASCII koda | HEX koda |
|-----------------------|---------------------|----------------------|-----------------------------|--|
| Zaključne kode | CR | Konec vrstice | 13 | 0x0d |
| | LF | Začetek nove vrstice | 10 | 0x0a |
| Podatki | 0–9 | Numerični podatki | 48–57 | 0x30–0x39 |
| | - (Minus) | Minus znak | 45 | 0x2d |
| | . (Decimalka) | Decimalna pika | 46 | 0x2e |
| | Presledek | Prazen prostor | 32 | 0x20 |
| | Napaka (Err 8.4) | Napaka | 69, 114, 114, 56, 46, 52 | 0x45, 0x72, 0x72, 0x38, 0x2e, 0x34 |
| | Tara | Tara mase | 78, 69, 84 | 0x4e, 0x45, 0x54 |
| | kg | Kilogram | 107, 103 | 0x6B, 0x67 |
| | g | Gram | 103 | 0x67 |
| | lb | Funt | 108, 98 | 0x6c, 0x62 |

| | | | | |
|-------|------------------|--|------------|------------------|
| | t | Tona | 116 | 0x74 |
| Ukazi | ON | Vklop tehtnice | 79, 78 | 0x4f, 0x4e |
| | OFF | Izklop tehtnice | 79, 70, 70 | 0x4f, 0x46, 0x46 |
| | IP | Takojšen izpis prikazane mase | 73, 80 | 0x49, 0x50 |
| | P | Izpis prikazane mase | 80 | 0x50 |
| | SP | Izpis stabilne mase | 83, 80 | 0x53, 0x50 |
| | CP | Kontinuirano izpisovanje mase | 67, 80 | 0x43, 0x50 |
| | XP | Izpis mase na intervalu x = 1 do 3600 sekund | 88, 80 | 0x58, 0x50 |
| | Z | Enakovredno pritisku na gumb ZERO | 90 | 0x5a |
| | T | Enakovredno pritisku na gumb TARE | 84 | 0x54 |
| | xT | Pred nastavljanje tare | 120, 84 | 0x78, 0x54 |
| | PU | Izpis trenutne merske enote | 80, 85 | 0x50, 0x55 |
| | xU | Spreminjanje merske enote | 120, 85 | 0x78, 0x55 |
| | PV | Izpis verzije programa | 80, 86 | 0x50, 0x56 |
| | H x »text« | Vnos glave | 72 | 0x48 |
| | Escape key and R | Ponastavitev na tovarniške nastavitve | 39, 82 | 0x27, 0x52 |

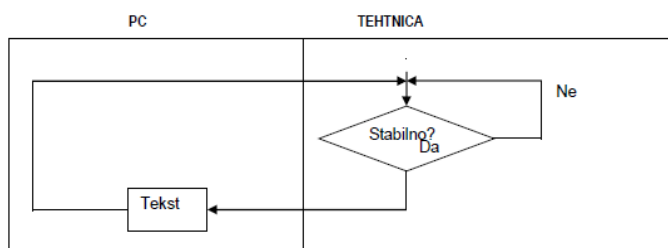
Tabela 6: Kontrolne kode in znaki, ki jih tehtnica pošilja na računalnik ali sprejema od računalnika.

3.2.2 Metode komuniciranja

Z nastavitvijo specifikacij na tehtnice lahko izbiramo med naslednjimi metodami komunikacije: tekoča (stalna), ročna in ukazna.

3.2.2.1 Tekoča (stalen prenos)

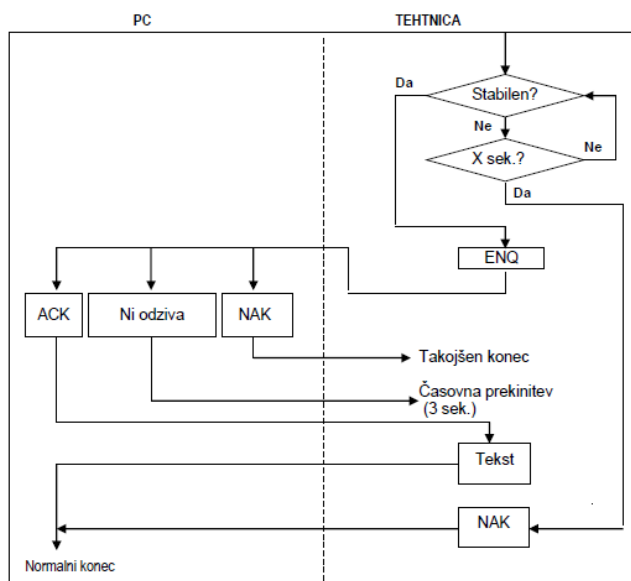
Podatki o masi se stalno oziroma kontinuirano prenašajo na računalnik (Slika 17). Preverjanje stabilnosti mase je odvisno od nastavitve specifikacij tehtnice (»stable print OFF« in »stable print ON«).



Slika 17: Stalen prenos podatka o masi na računalnik.

3.2.2.2 Ročna

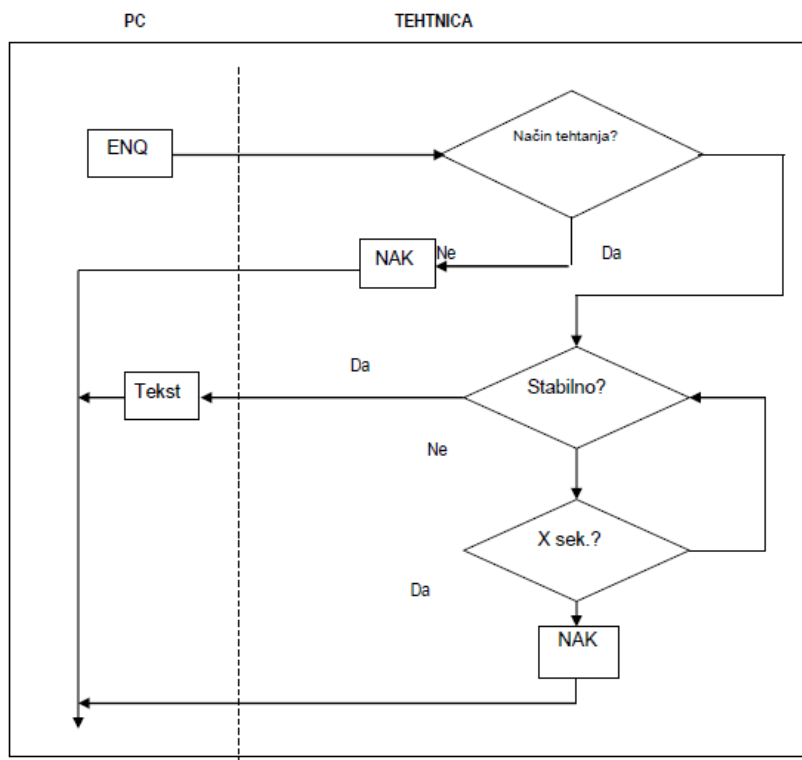
Podatki o masi se prenesejo ob pritisku na tipko »P-rint« (Slika 18). Uporabnik lahko z nastavitvijo specifikacij na tehtnici izbere, ali se podatki prenesejo takoj ali pa se ukaz zadrži, dokler masa ne postane stabilna. Preverjanje stabilnosti mase je odvisno od nastavitve specifikacij tehtnice (»stable print OFF« in »stable print ON«).



Slika 18: Ročni prenos podatkov o masi na računalnik.

3.2.2.3 Ukazna

Prenos podatkov o masi se začne s sprejemom ukaza od zunanje naprave (na primer računalnika (Slika 19)). Računalnik pošlje tehtnici ukaz za izpis prikazane mase na digitalnem indikatorju tehtnice (ASCII 80). Preverjanje stabilnosti mase je odvisno od nastavitve specifikacij tehtnice (»stable print OFF« in »stable print ON«).



Slika 19: Ukazen prenos podatka o masi na računalnik.

Za potrebe naše aplikacije smo uporabili serijski komunikacijski protokol RS-232 z naslednjimi parametri povezave: **hitrost prenosa 9600 bps, 8 podatkovnih bitov, brez parnost, en stop bit** (Slika 16). Uporabili smo metodo **tekoč (stalen) prenos s preverjanjem stabilnosti mase** (Slika 17). Tehtnica po tej metodi kontinuirano pošilja podatke o masi na računalnik, in sicer, ko tehtnica zabeleži tri zaporedne identične meritve pozitivne mase v intervalnem zamiku ene sekunde, se podatki o masi pošljejo na računalnik.

Poglavje 4 **Izdelava aplikacije**

V prejšnjem poglavju smo opisali serijski vmesnik RS-232, preko katerega je tehtnica povezana z računalnikom, kot tudi protokol, s katerim se lahko računalnik preko serijskega vmesnika sporazumeva s tehtnico.

V tem poglavju pa bomo opisali uporabniške zahteve aplikacije, funkcionalnosti aplikacije, podatkovno bazo, ki jo aplikacija uporablja, in uporabniški vmesnik aplikacije.

4.1 Opis aplikacije

Programska oprema ATS.NET je aplikacija, ki omogoča povezavo s cestno tehtnico Ohaus (v nadaljevanju: tehtnica).

Aplikacija je namenjena podjetjem, ki uporabljajo tehtnico in želijo beležiti vhodna in izhodna tehtanja tovornih vozil, jih arhivirati/shranjevati v računalniku, izvažati podatke tehtanj v CSV datoteko in jih na-to uporabljati za nadaljnjo uporabo, na primer v lastnih aplikacijah ali za obdelavo v Excelu.

Aplikacija nam omogoča sprejemanje mase na dva načina (odvisno, kako je nastavljena tehtnica):

- za sprejemanje mase iz tehtnice je potreben pritisk tipke »P-print« oziroma druge pred nastavljene tipke na tehtnici, ki omogoča prenos mase iz tehtnice v računalnik;
- stalen (kontinuiran) prenos – tehtnica ves čas pošilja maso v računalnik.

Aplikacija nudi podatkovno bazo, kamor se shranjujejo vsi podatki o tehtanjih, podjetjih, vozilih, materialih in uporabnikih. Podatkovna baza je namenjena pohitritvi izvajanja tehtanj in shranjevanju le-teh po ključih, na podlagi katerih lahko izdelujemo poročila, vodimo zaloge, pregledujemo statistiko in izdajamo tehtalne liste.

Aplikacija nudi možnost izpisa podatkovnih tabel ter izvoza le-teh v datoteko CSV. Omogočen je pregled vseh opravljenih tehtanj z možnostjo filtriranja po kriterijih (datum od/do, številka tehtalnega lista, naziv podjetja, material in registrska številka vozila). Vsa opravljena tehtanja

je možno natisniti in izvoziti v datoteko CSV. Aplikacija omogoča pregled statistike po materialih in podjetjih z možnostjo datumske omejitve ter grafično ponazoritev dnevne statistike pripeljanega in odpeljanega materiala. Aplikacija uporabniku omogoča funkciji tariranja in ničliranja tehtnice iz računalnika.

Dostop do aplikacije je omejen z uporabniškimi imeni in gesli. Na ta način lahko spremljamo, kdo je opravljajal katera tehtanja.

Programska oprema je univerzalna, kar pomeni, da ob menjavi tehtnice ne bo treba menjati aplikacije, temveč samo spremenimo protokol tehtnice ter delamo naprej z isto programsko opremo. S tem se izognemo izgubi tehtanj ali migraciji podatkov in ponovnemu uvajanju operaterjev za delo z novo aplikacijo.

4.2 Uporabniške zahteve

Uporabniške zahteve opisujejo funkcionalnosti aplikacije, ki bodo na voljo končnemu uporabniku. Vsaka uporabniška zahteva predstavlja funkcionalnost, ki bo za uporabnika koristna. Pri razvoju aplikacije smo se držali podanih uporabniških zahtev. V nadaljevanju so opisane splošne zahteve uporabnika in glavne funkcionalnosti aplikacije.

Splošne zahteve:

- izdelati je treba aplikacijo, s katero bo mogoče beleženje vhodnih in izhodnih tehtanj tovornih vozil;
- aplikacija mora delovati na Microsoftovih operacijskih sistemih;
- aplikacija mora za shranjevanje podatkov uporabljati podatkovno bazo MySQL;
- dostop do aplikacije mora biti varovan z uporabniškim imenom in geslom;
- aplikacija mora biti skalabilna za dodajanje novih funkcionalnosti v prihodnosti;
- aplikacija mora biti enostavna za uporabo.

Glavne funkcionalnosti:

- aplikacija mora avtomatično zajemati podatke o masi iz tehtnice;

- aplikacija mora omogočati nastavitve oziroma spreminjanje parametrov za serijsko komunikacijo po standardu RS-232 in avtomatično prepoznati, na kateri serijski vmesnik oziroma serijska vrata (angl. com port) je priključena tehtnica;
- aplikacija mora omogočati opravljanje tehtanj (vhodno tehtanje in izhodno tehtanje);
- aplikacija mora omogočati vnos novega podjetja, spreminjanje podatkov obstoječega podjetja, brisanje podjetja, izvoz podatkov o podjetju v CSV datoteko;
- aplikacija mora omogočati vnos novega materiala, spreminjanje podatkov obstoječega materiala, brisanje materiala, izvoz podatkov o materialu v CSV datoteko;
- aplikacija mora omogočati vnos novega vozila, spreminjanje podatkov obstoječega vozila, brisanje vozila, izvoz podatkov o vozilu v CSV datoteko;
- aplikacija mora omogočati vnos novega uporabnika, spreminjanje podatkov obstoječega uporabnika, brisanje uporabnika in določitev vloge oziroma tipa uporabnika (administrator/končni uporabnik);
- aplikacija mora omogočati pregled vseh opravljenih tehtanj (zgodovina tehtanj) z možnostjo filtriranja po kriterijih (datum od/do, številka tehtalnega lista, naziv podjetja, material in registrska številka vozila);
- aplikacija mora omogočati pregled vseh odprtih tehtanj;
- aplikacija mora omogočati pregled statistike:
 - po materialih: pregled količine pripeljanega in odpeljanega materiala v izbranem časovnem obdobju;
 - po podjetjih: pregled količine pripeljanega in odpeljanega materiala v izbranem časovnem obdobju za vsa podjetja;
 - grafična ponazoritev dnevne količine pripeljanega in odpeljanega materiala;
- aplikacija mora omogočati tiskanje tehtalnega lista za vhodno in izhodno tehtanje;
- aplikacija mora omogočati tariranje in ničliranje tehtnice iz računalnika:
 - ničliranje je postopek, ko se po vklopu ali ob pritisku na tipko za ničliranje tehtnica postavi v izhodiščni položaj »0«;

- funkcija tara služi posebej za merjenje snovi, kot so tekočine, praški itd. (snovi, ki niso trdne). Funkcija tara izravna težo vozila, tako da se na zaslonu tehtnice prikaže le neto teža snovi, ki jo želimo stehtati.

4.2.1 Primer uporabe

Z modelom primerov uporabe (angl. use case diagram) [21] določimo načine komuniciranja uporabnikov s sistemom, hkrati pa določimo komunikacijo med primeri uporabe v sistemu. Primeri uporabe so običajno namenjeni zajemanju in predstavitvi uporabniških zahtev. Sestavljen je iz grafične predstavitve primerov uporabe in opisa scenarijev primerov uporabe v obliki besedila.

Osnovne komponente modela primerov uporabe [21]:

- akter;
- primer uporabe;
- povezava med akterjem in primerom uporabe;
- povezava med dvema primeroma uporabe;
- povezava med dvema akterjema;
- povezava med sistemom in podsistemom.

Povezave oziroma relacije, ki med seboj povezujejo primere uporabe:

- relacija vsebovanja (angl. include): vključuje funkcionalnost;
- relacija razširja (angl. extend): razširja obnašanje;
- relacija posploševanja (angl. generalization): več akterjev posplošimo v enega.

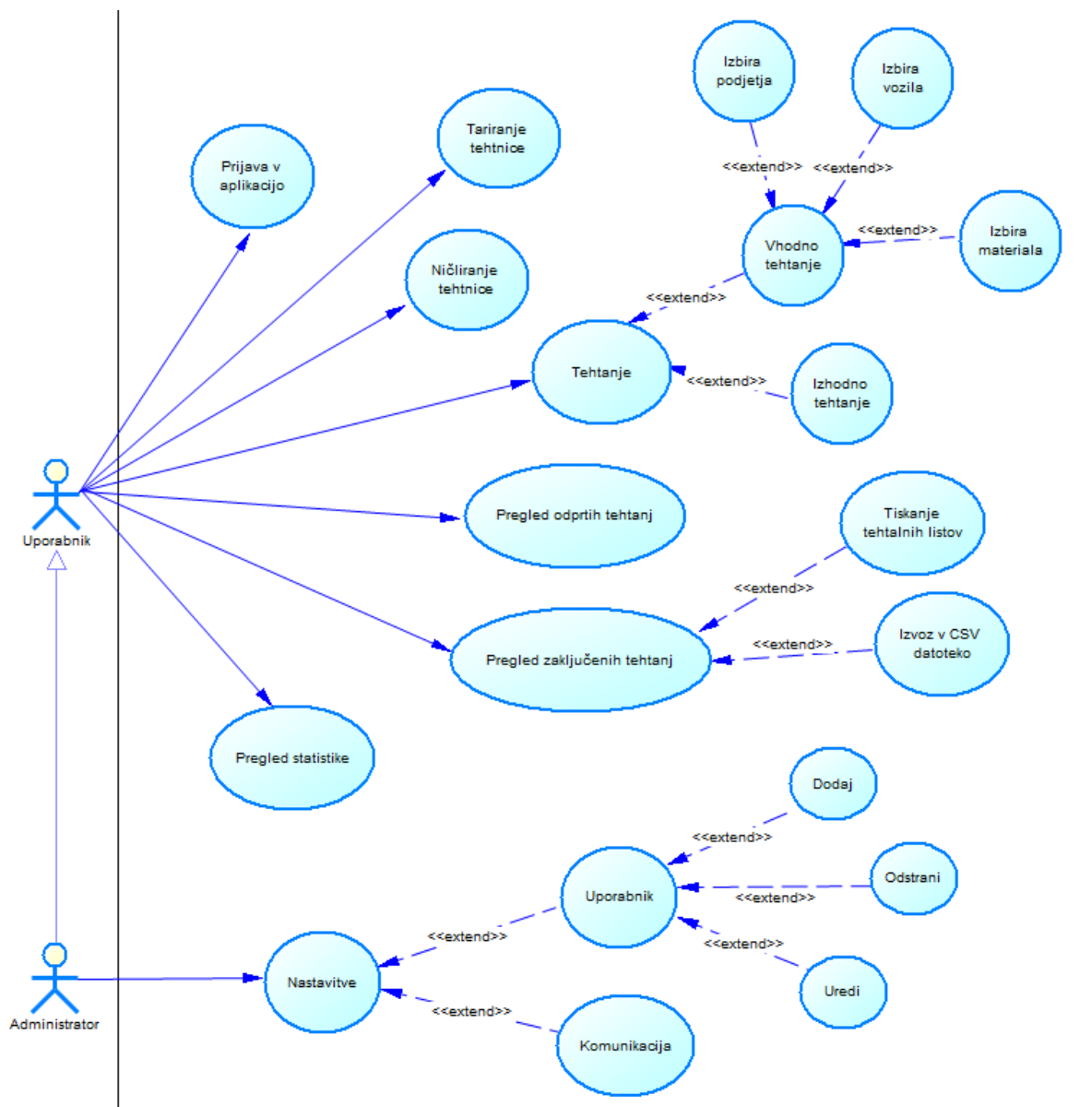
Primere uporabe lahko uporabljamo tudi za [21]: ocenjevanje velikosti projekta in potrebnih virov, komunikacijo s končnim uporabnikom in strankami, identificiranje objektov v sistemu, določanje funkcionalnosti objektov, načrtovanje uporabniških vmesnikov, definiranje interakcije med objekti in vmesniki objektov, testiranje – najpogosteje v prvih fazah objektnega načrtovanja aplikacij.

Diagram primerov uporabe smo uporabili za prikaz in lažje razumevanje uporabniških zahtev in funkcionalnosti aplikacije (Slika 20).

Uporabnik ima naslednje funkcionalnosti:

- prijava v aplikacijo;
- opravljanje tehtanj (vhodno tehtanje in izhodno tehtanje);
- izbira podjetja (vnos/spreminjanje/brisanje podjetij);
- izbira materialov (vnos/spreminjanje/brisanje materialov);
- izbira vozila (vnos/spreminjanje/brisanje vozil);
- pregled vseh opravljenih oziroma zaključenih tehtanj;
- pregled vseh odprtih tehtanj;
- tiskanje tehtalnih listov opravljenih oziroma zaključenih tehtanj;
- izvoz opravljenih oziroma zaključenih tehtanj v CSV datoteko (Excel);
- tariranje in ničliranje tehtnice;
- pregled statistike.

Administrator generalizira uporabnika, kar pomeni, da ima poleg njegovih funkcionalnosti možnost nastavljanja komunikacije s tehtnico in upravljanje s končnimi uporabniki (dodajanje/spreminjanje/brisanje uporabnikov).



Slika 20: Diagram primerov uporabe za celoten sistem.

4.3 Podatkovna baza

Podatkovna baza [22] je zbirka podatkov oziroma dokumentov, ki so zapisani v posameznih tabelah podatkovne zbirke. Zbirke se lahko medsebojno kličejo, razvrščajo, iščejo in urejajo podatke ter služijo kot osnova za sprejemanje odločitev in izvajanje akcij in zahtev.

Izdelavo aplikacije smo začeli z načrtovanjem podatkovne baze. Za hranjenje podatkov smo izbrali prosto dostopno podatkovno bazo MySQL (razdelek 2.2.3). Za podatkovno bazo MySQL smo se odločili, ker je odprtokodna, enostavna za namestitev, dobro dokumentirana, predvsem pa zadostuje potrebam naše aplikacije, saj vsebuje vse pomembne elemente podatkovnih baz.

Podatkovno bazo smo izdelali s pomočjo orodja MySQL Workbench (razdelek 2.2.2). Zanj smo se odločili, ker je prosto dostopno, enostavno za uporabo in namestitev. Orodje ponuja funkcijo vnaprejšnjega inženiringa podatkovne baze (angl. Forward Engineering to Databases), s katero lahko podatkovni model (Slika 18) pretvorimo v podatkovno bazo. MySQL Workbench nam izdelava vse potrebne skripte, s pomočjo katerih nato ustvari relacijsko podatkovno bazo. Funkcija vnaprejšnjega inženiringa podatkovne baze nam iz podatkovnega modela zgradi tabele z ustreznimi lastnostmi (atributi tabele s pripadajočimi tipi, primarni in tuji ključi, povezave med tabelami itd.).

4.3.1 Podatkovni model

Podatkovni model [23] je povezana zbirka konceptov, namenjenih opisovanju in manipulaciji s podatki. Podatkovne modele uporabljamo za predstavitev konceptualnega in zunanjega nivoja podatkovne baze. Podatkovni model opisuje okolje s termini, kot so entiteta, atribut, relacija in ključ.

Na podlagi uporabniških zahtev, ki uporabniku omogočajo izvajanje vhodnih in izhodnih tehtanj tovornih vozil, smo analizirali celoten problem, definirali, katere entitete so v aplikaciji potrebne, narisali entitetno-relacijski diagram podatkovnega modela in ga nato implementirali na strežnik MySQL. Za izdelavo in implementacijo podatkovnega modela smo uporabili orodje MySQL Workbench.

V orodju MySQL Workbench podatkovni model izdelamo tako, da v zavihku »File« izberemo »New Model«. V oknu, ki se nam odpre, izberemo »Add new Diagram«. V oknu »Diagram« narišemo entitetno-relacijski diagram podatkovnega modela in vanj vnesemo vse potrebne podatke (attribute tabele s pripadajočimi tipi, povezave med tabelami, primarne in tuje ključe itd.).

Izdelan model vsebuje 5 entitet, kamor se bodo shranjevali vsi potrebni podatki.

V aplikaciji so definirane naslednje entitete.

Uporabniki: vsebovala bo podatke o posameznem uporabniku. Tabela uporabnikov je namenjena identifikaciji osebe, ki izvaja tehtanje. Uporabniki aplikacije so lahko administratorji ali pa končni uporabniki:

- idUporabnika (INT) – zaporedna številka zapisa, primarni ključ;
- ime (VARCHAR) – ime uporabnika;
- priimek (VARCHAR) – priimek uporabnika;
- uporabniškoIme (VARCHAR) – uporabniško ime uporabnika;
- geslo (VARCHAR) – geslo uporabnika;
- vloga (VARCHAR) – vloga uporabnika (administrator, uporabnik).

Podjetja: vsebovala bo podatke o posameznem podjetju. Tabela podjetij je namenjena temu, da redne stranke in dobavitelje vnesemo in jih kasneje samo izbiramo, ne da bi pri tem morali vsakič znova vnašati vse podatke podjetja, kot so naziv, naslov itd.:

- idPodjetja (INT) – zaporedna številka zapisa, primarni ključ;
- naziv (VARCHAR) – naziv podjetja;
- naslov (VARCHAR) – naslov podjetja;
- mesto (VARCHAR) – mesto v katerem se podjetje nahaja;
- telefon (VARCHAR) – telefon podjetja;
- email (VARCHAR) – email podjetja.

Vozila: vsebovala bo podatke o posameznem vozilu. Tabela vozil je namenjena temu, da vsa vozila, ki redno vozijo v podjetje, vnesemo v podatkovno bazo in jih kasneje le izbiramo ter dodeljujemo posameznim podjetjem:

- idVozila (INT) – zaporedna številka zapisa, primarni ključ;
- idPodjetja (INT) – zaporedna številka zapisa v tabeli Podjetja, tuji ključ;

- registrskaŠtevilka (VARCHAR) – registrska številka vozila;
- opis (VARCHAR) – opis vozila;
- voznik (VARCHAR) – voznik vozila;
- največjaDovoljenaMasa (DOUBLE) – največja dovoljena masa vozila.

Materiali: vsebovala bo podatke o posameznem materialu. Tabela materialov je namenjena temu, da vse materiale, ki se redno tehtajo, vnesemo v podatkovno bazo in jih kasneje le izbiramo ter dodeljujemo posameznim podjetjem:

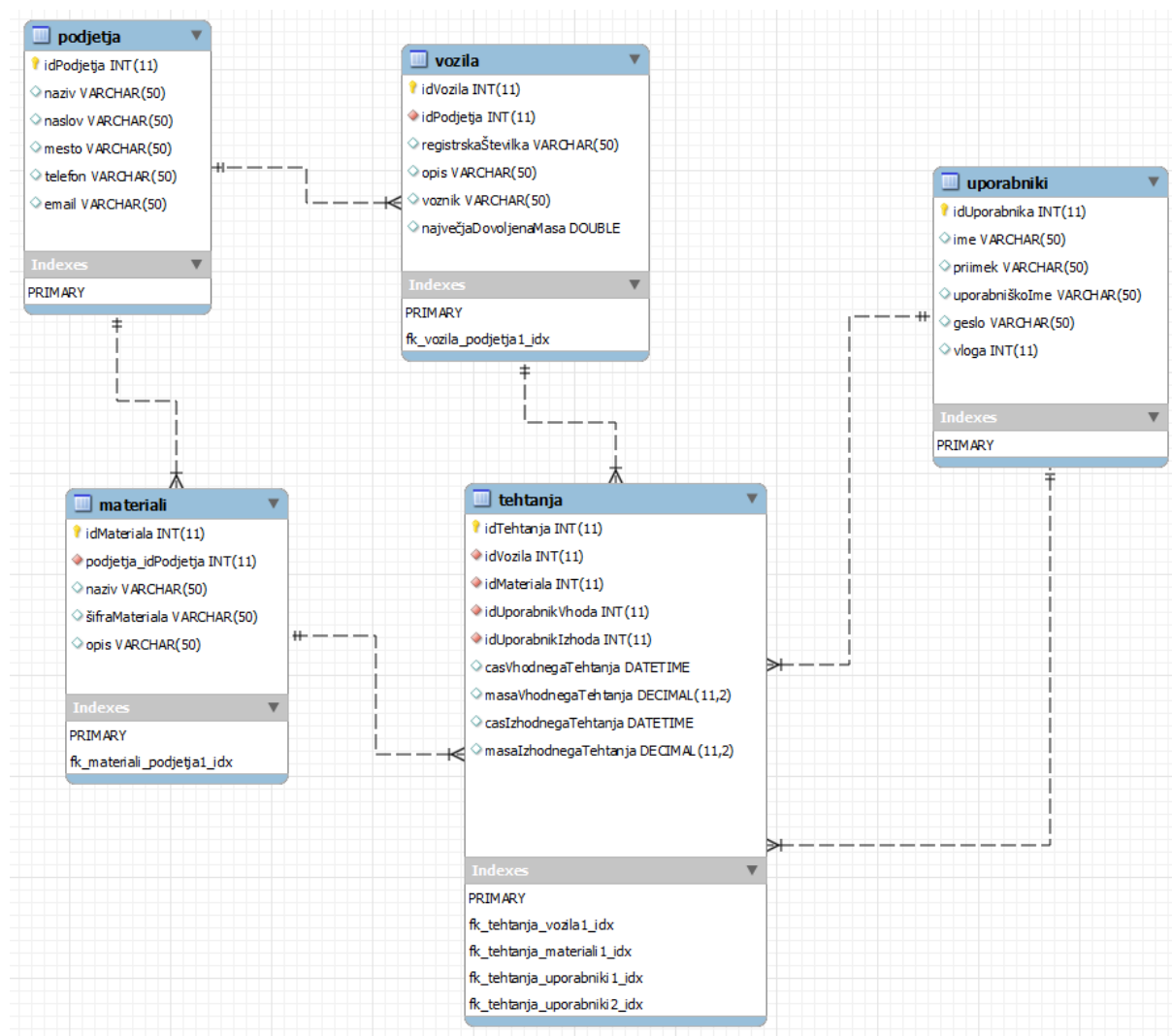
- idMateriala (INT) – zaporedna številka zapisa, primarni ključ;
- idPodjetja (INT) – zaporedna številka zapisa v tabeli Podjetja, tuji ključ;
- naziv (VARCHAR) – naziv materiala;
- šifraMateriala (VARCHAR) – šifra materiala;
- opis (VARCHAR) – opis materiala.

Tehtanja: vsebovala bo podatke o posameznem tehtanju vozila. Tabela tehtanj služi shranjevanju vseh opravljenih tehtanj (zgodovina tehtanj). V tej tabeli so shranjena le zaključena tehtanja, tj. brez odprtih tehtanj;

- idTehtanja (INT) – zaporedna številka zapisa, primarni ključ;
- idVozila (INT) – zaporedna številka zapisa v tabeli Vozila, tuji ključ;
- idMateriala (INT) – zaporedna številka zapisa v tabeli Materiali, tuji ključ;
- casVhodnegaTehtanja (DATETIME) – čas vhodnjega tehtanja vozila;
- masaVhodnegaTehtanja (DECIMAL) – masa vhodnega tehtanja vozila;
- idUporabnikVhoda (INT) – zaporedna številka zapisa v tabeli Uporabniki, tuji ključ;
- casIzhodnegaTehtanja (DATETIME) – čas izhodnega tehtanja vozila;
- masaIzhodnegaTehtanja (DECIMAL) – masa izhodnega tehtanja vozila;
- idUporabnikIzhoda (INT) – zaporedna številka zapisa v tabeli Uporabniki, tuji ključ.

Entitete so med seboj povezane z relacijo 1 : N, kar pomeni, da se zapis v prvi entiteti lahko poljubnokrat uporabi v drugi entiteti. Primer: eno podjetje ima lahko več vozil, medtem ko eno vozilo pripada le enemu podjetju.

Slika 21 prikazuje diagram podatkovnega modela in relacije med posameznimi entitetami.



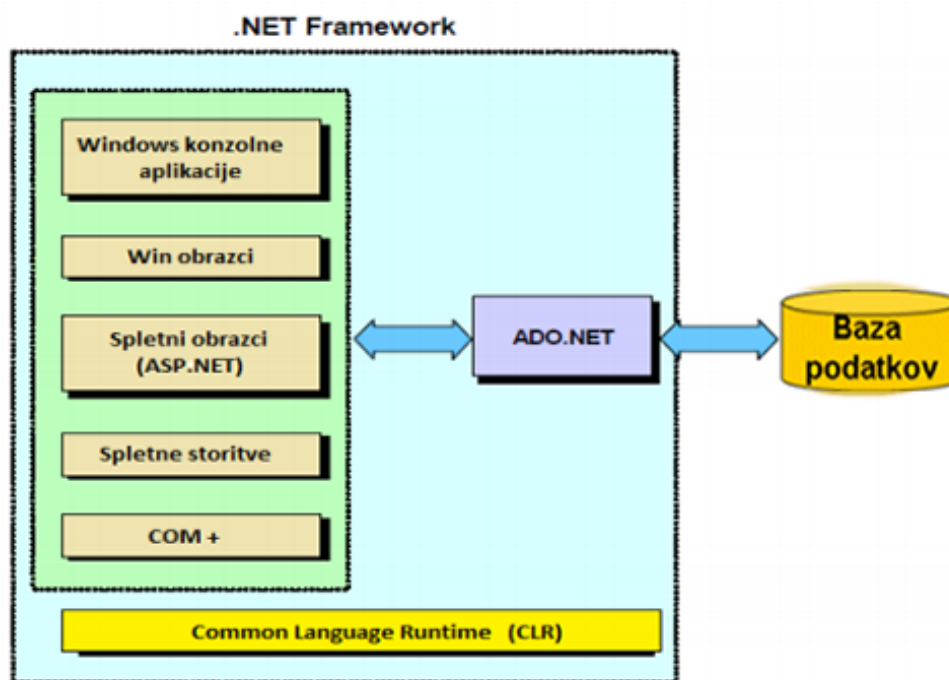
Slika 21: Podatkovni model naše podatkovne baze.

4.3.2 Dostop do podatkovne baze

Aplikaciji smo dostop do podatkovne baze omogočili s pomočjo orodja Microsoft Visual Studio. Aplikacija dostopa do podatkovne baze preko posebne knjižnice MySQL Connector/NET, ki omogoča .NET aplikacijam komunikacijo s strežnikom MySQL. MySQL Connector/NET je na voljo na spletnem naslovu <http://dev.mysql.com/downloads/connector/net/>.

4.3.2.1 MySQL Connector/NET (.NET connection)

Connector/NET [24] nam omogoča enostavno izdelavo .NET aplikacij, ki zahtevajo varno in visoko zmogljivo povezljivost s podatkovno bazo MySQL. Implementira potrebne vmesnike ADO.NET in integrira orodja ADO.NET. Razvijalci lahko za izdelavo aplikacije uporabljajo kateri koli .NET jezik. Slika 22 prikazuje delovanje ADO.NET-a.



Slika 22: Uporaba ADO.NET-a.

Connector/NET vključuje popolno podporo za [24]:

- funkcionalnosti, ki jih omogoča strežnik MySQL;
- protokol za stiskanje, ki omogoča stiskanje podatkovnega toka med strežnikom in odjemalcem;
- povezavo z uporabo vtičnice TCP/IP;

- odprtokodno ogrodje Mono;
- Microsoft Entity Framework;
- Microsoft Windows RT.

Knjižnica za povezovanje na podatkovno bazo MySQL in ustvarjanje ustreznih poizvedb v .NET okolju je vsebovana v datoteki MySql.Data.dll, ki jo dodamo v naš projekt. Za povezavo potrebujemo povezovalni niz, v katerem navedemo vse potrebne podatke.

Niz: `private string myConnectionString = " server=192.1.198.200; user id=tehtanjeAdmin; password=*****; database=tehtnica ";`

- **server** – naš strežnik MySql;
- **user id** – naše uporabniško ime;
- **password** – naše geslo za strežnik;
- **database** – ime naše podatkovne baze, v kateri imamo podatke.

4.3.2.2 Izgradnja povezave za dostop do podatkovne baze MySQL

V Tabeli 7 je prikazan primer izgradnje povezave za dostop do podatkovne baze MySQL. Povezava se vzpostavi s pomočjo gradnika Connection (MySqlConnection). Gradnik Connection skrbi za povezavo s podatkovno bazo MySQL. Njegove metode in lastnosti (Open(), Close(), User, Password, itd.) zagotavljajo nadzor in vzdrževanje povezave. Pomembna lastnost objekta tega tipa je ConnectionString, kjer je točno naveden podatkovni vir, s katerim bo objekt povezan. Gradniki Command (MySqlCommand) omogočajo izvajanje različnih ukazov SQL ali shranjenih procedur (angl. stored procedures) nad aktivno povezavo do podatkovne baze MySQL. Poznamo štiri lastnosti tipa Command:

- SelectCommand omogoča prebiranje podatkov iz podatkovne baze MySQL;
- InsertCommand omogoča dodajanje novih vrednosti v podatkovno bazo MySQL;
- UpdateCommand omogoča spreminjanje že obstoječih vrednosti v podatkovni bazi MySQL;
- DeleteCommand omogoča brisanje ustreznih podatkov iz podatkovne baze MySQL.

S pomočjo gradnika DataReader (MySqlDataReader) se lahko dostopa oziroma pridobiva podatke iz podatkovne baze MySQL.

```
using MySql.Data.MySqlClient;

// V povezovalnem nizu myConnectionString navedemo lokacijo podatkovne baze, ime
// podatkovne baze, uporabniško ime in geslo.

string myConnectionString = "server=192.1.198.200; user id=tehtanjeAdmin;
password=*****; database=tehtnica";

// Vsebina poizvedbe.

string Query = " SELECT *
                FROM tehtnica.uporabnik
                WHERE UporabniskoIme= '" + this.uporabnisko_ime.Text + "'
                AND Geslo= '" + this.geslo.Text + "';";

// Ustvarimo objekt tipa MySqlConnection za povezavo s podatkovno bazo.
// Konstruktor objekta tipa MySqlConnection dobi podatke o izvoru podatkovne baze.

MySqlConnection conDataBase = new MySqlConnection(myConnectionString);

// Ustavrimo objekt tipa MySqlCommand, ki vzpostavi komunikacijo s podatkovno bazo
// in napove vrsto interakcije s podatkovno bazo.

MySqlCommand SelectCommand = new MySqlCommand (Query, conDataBase);

// Odpremo povezavo s podatkovno bazo.

conDataBase.Open();

// Z razredom MySqlDataReader ustvarimo podatkovni tok za branje vrstic iz tabele v
// podatkovni bazi.

MySqlDataReader myReader = SelectCommand.ExecuteReader();

    // Izvedba poizvedbe.

// Zapremo podatkovni tok.

myReader.Close();

// Zapremo povezavo s podatkovno bazo.

conDataBase.Close();
```

Tabela 7: Primer izgradnje povezave za dostop do podatkovne baze MySQL.

4.3.2.3 Primer dodajanja uporabnika v podatkovno bazo MySQL

Kot primer povezovanja med aplikacijo in podatkovno bazo si v Tabeli 8 oglejmo, kako v naši aplikaciji poteka dodajanje uporabnika v podatkovno bazo MySQL.

```
private void btnDodajUporabnika_Click(object sender, EventArgs e){

    // V povezovalnem nizu myConnectionString navedemo lokacijo baze, ime baze,
    uporabniško ime in geslo.

    string myConnectionString = "server=192.1.198.200; user id=tehtanjeAdmin;
    password=*****; database=tehtnica";

    // Vsebina poizvedbe.

    string Query = "INSERT into tehtnica.uporabnik(Ime, Priimek, UporabniskoIme, Geslo,
    Vloga) values ('" + this.ime_txt.Text + "' , '" + this.priimek_txt.Text + "' , '" +
    this.up_ime_txt.Text + "' , '" + this.geslo_txt.Text + "', '" +s+ "') ";

    // Ustvarimo objekt tipa MySqlConnection za povezavo s podatkovno bazo.
    Konstruktor objekta tipa MySqlConnection dobi podatke o izvoru podatkovne baze.

    MySqlConnection conDataBase = new MySqlConnection(myConnectionString);

    // Ustavrimo objekt tipa MySqlCommand, ki vzpostavi komunikacijo s podatkovno bazo
    in napove vrsto interakcije s podatkovno bazo.

    MySqlCommand InsertCommand = new MySqlCommand(Query, conDataBase);

    // Odpremo povezavo s podatkovno bazo s pomočjo try/catch bloka.

    try{
        conDataBase.Open();

        // Z metodo BeginExecuteNonQuery zapišemo podatke o uporabniku v tabelo.

        InsertCommand.ExecuteNonQuery();
        MessageBox.Show("Dodajanje uporabnika je uspešno!");

        // Zapremo povezavo s podatkovno bazo.

        conDataBase.Close();

    }

    catch (Exception ex){
        MessageBox.Show("Napaka pri shranjevanju podatkov!");
    }

    nastavitve.load_table();

}
```

Tabela 8: Primer, ki prikazuje dodajanje uporabnika v podatkovno bazo MySQL.

4.3.2.4 Primer prikaza podatkov o uporabnikih, ki jih pridobimo iz podatkovne baze MySQL

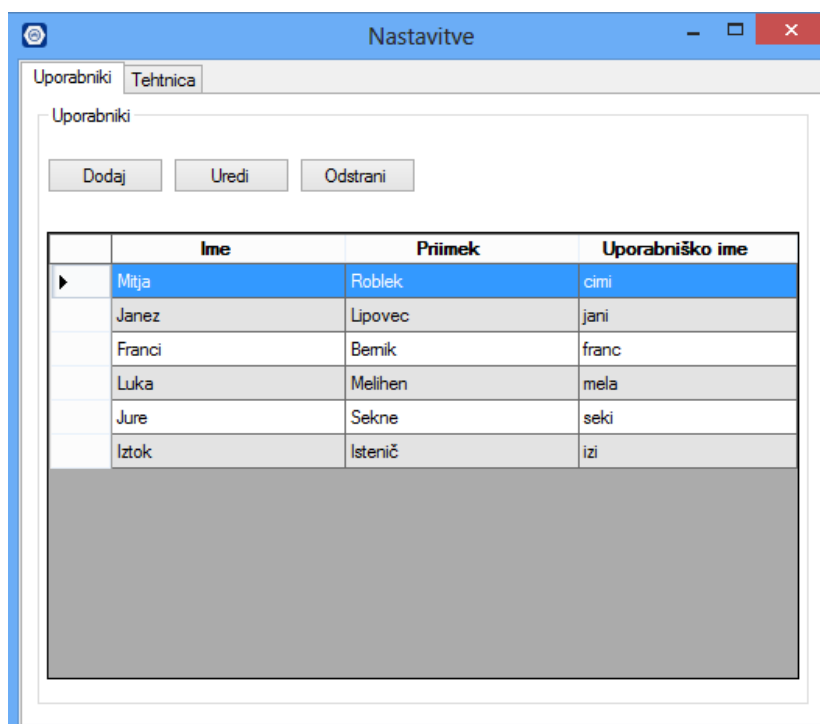
Kot primer povezovanja med aplikacijo in podatkovno bazo si v Tabeli 9 oglejmo še, kako v naši aplikaciji poteka prikaz podatkov o uporabnikih, ki jih pridobimo iz podatkovne baze MySQL. Vsebino tabele »Uporabnik« iz podatkovne baze »Tehtnica« bomo prikazali v gradniku »DataGridView« (Slika 23).

```
public void load_table(){  
  
    // V povezovalnem nizu myConnectionString navedemo lokacijo baze, ime baze,  
    uporabniško ime in geslo.  
  
    string myConnectionString = "server=192.1.198.200; user id=tehtanjeAdmin;  
    password=*****; database=tehtnica";  
  
    // Vsebina poizvedbe.  
  
    string Query = "SELECT idUporabnika as '#', Ime, Priimek, UporabniskoIme as  
    'Uporabniško ime', Vloga FROM tehtnica.uporabnik";  
  
    // Ustvarimo objekt tipa MySqlConnection za povezavo s podatkovno bazo.  
    Konstruktor objekta tipa MySqlConnection dobi podatke o izvoru podatkovne baze.  
  
    MySqlConnection conDataBase = new MySqlConnection(myConnectionString);  
  
    // Ustavrimo objekt tipa MySqlCommand, ki vzpostavi komunikacijo s podatkovno bazo  
    in napove vrsto interakcije s podatkovno bazo.  
  
    MySqlCommand cmdDataBase = new MySqlCommand(Query, conDataBase);  
  
    try{  
  
        // Vzpostavimo komunikacijo s podatkovno bazo in izvedemo poizvedbo.  
  
        MySqlDataAdapter sda = new MySqlDataAdapter();  
        sda.SelectCommand = cmdDataBase;  
  
        // Podatke iz tabele Uporabnik uvozimo v objekt tipa DataTable.  
  
        DataTable dbdataset = new DataTable();  
  
        // Objekt tabele napolnimo s podatki iz podatkovne baze.  
  
        sda.Fill(dbdataset);  
        BindingSource bSource = new BindingSource();  
        bSource.DataSource = dbdataset;  
  
        // Gradnik DataGridView povežemo s podatki gradnika DataTable.  
  
        dataGridView1.DataSource = bSource;  
        sda.Update(dbdataset);  
    }  
    catch (Exception ex){  
        MessageBox.Show(ex.Message);  
    }  
}
```

```
// Oblikovanje gradnika DataGridView v katerem prikažemo podatke.

dataGridView1.ColumnHeadersDefaultCellStyle.Font = new
System.Drawing.Font("Microsoft Sans Serif", 8, FontStyle.Bold, GraphicsUnit.Point);
dataGridView1.ColumnHeadersDefaultCellStyle.BackColor = SystemColors.ControlDark;
dataGridView1.ColumnHeadersBorderStyle = DataGridViewHeaderBorderStyle.Single;
dataGridView1.ColumnHeadersDefaultCellStyle.Alignment=
DataGridViewContentAlignment.MiddleCenter;
dataGridView1.DefaultCellStyle.Font = new System.Drawing.Font("Microsoft Sans
Serif", 8, FontStyle.Regular, GraphicsUnit.Point);
dataGridView1.DefaultCellStyle.BackColor = Color.Empty;
dataGridView1.AlternatingRowsDefaultCellStyle.BackColor = SystemColors.ControlLight;
dataGridView1.CellBorderStyle = DataGridViewCellBorderStyle.Single;
dataGridView1.GridColor = SystemColors.ControlDarkDark;
this.dataGridView1.AutoSizeColumnsMode=
DataGridViewAutoSizeColumnsMode.DisplayedCells;
this.dataGridView1.Columns["#"].Visible = false;
this.dataGridView1.Columns[1].AutoSizeMode = DataGridViewAutoSizeMode.None;
this.dataGridView1.Columns[1].Width = 150;
this.dataGridView1.Columns[2].AutoSizeMode = DataGridViewAutoSizeMode.None;
this.dataGridView1.Columns[2].Width = 150;
this.dataGridView1.Columns[3].AutoSizeMode = DataGridViewAutoSizeMode.Fill;
this.dataGridView1.AutoSizeColumnsMode=
DataGridViewAutoSizeColumnsMode.DisplayedCells;
this.dataGridView1.Columns[4].Visible = false;
}
```

Tabela 9: Primer prikaza podatkov o uporabnikih, ki jih pridobimo iz podatkovne baze MySQL.



Slika 23: Povezava gradnika »DataGridView« s tabelo »Uporabniki«.

4.4 Uporabniški vmesnik

Uporabniški vmesnik [25] (angl. user interface, UI) je okolje, v katerem uporabnik komunicira z računalnikom oziroma napravo. V uporabi so predvsem grafični uporabniški vmesniki (angl. graphical user interface, GUI) in vmesniki z ukazno vrstico (angl. command line interface, CLI). Grafični uporabniški vmesnik je vmesnik, ki uporablja okna, ikone, menije in druge grafične elemente. Uporabljamo ga s pomočjo miške, tipkovnice ali elektronskega pisala. Za grafični uporabniški vmesnik lahko uporabljamo tudi kratico WIMP (angl. windows, icons, menus, pointers).

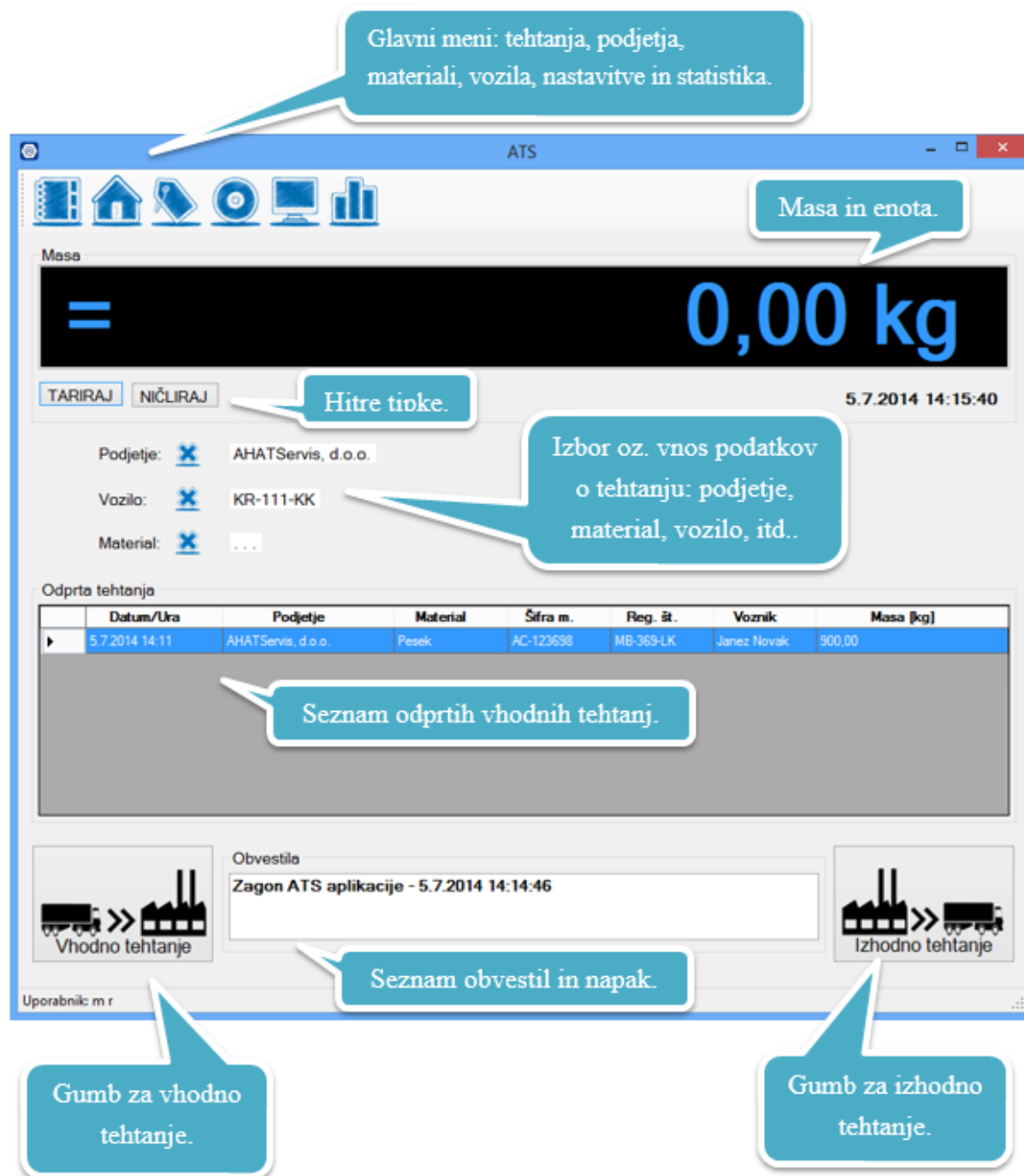
Načrtovanje uporabniškega vmesnika je pomembna aktivnost, saj uporabniki kvaliteto aplikacije velikokrat enačijo s kvaliteto oziroma izgledom uporabniškega vmesnika. Pravilno oblikovan uporabniški vmesnik je zato pomemben del vsake aplikacije [26]. Slika 24 prikazuje sestavne dele uporabniškega vmesnika.



Slika 24: Sestavni deli uporabniškega vmesnika.

Glede na podane uporabniške zahteve je bilo treba oblikovati in izdelati grafični uporabniški vmesnik, ki v obliki ikon in drugih grafičnih elementov predstavlja funkcije in možnosti, ki so na voljo uporabniku. Grafični uporabniški vmesnik smo oblikovali in izdelali s pomočjo orodja Microsoft Visual Studio, ki vsebuje vse potrebne razrede in komponente za izdelavo okenskih aplikacij v okolju Windows. Temelji na tehnologiji .NET. Sestavljen je iz več oken, od katerih vsako služi svojemu namenu. Obsega okna za prijavo v sistem, pregled zaključenih tehtanj, izbiro podjetja, izbiro materiala, izbiro vozila, nastavitve, pregled statistike, iskanje podjetja, iskanje vozila, iskanje materiala, dodajanje podjetja, urejanje podjetja, dodajanje vozila, urejanje vozila, dodajanje materiala, urejanje materiala, dodajanje uporabnika, urejanje uporabnika in glavno okno aplikacije, preko katerega lahko opravljamo vhodna in izhodna tehtanja in dostopamo do prej naštetih oken.

Uporabniški vmesnik smo zasnovali tako, da je čim bolj pregleden (veliki gumbi/ikone nakazujejo svojo funkcionalnost) in omogoča hiter in enostaven dostop do želenih funkcionalnosti in podatkov. Slika 25 prikazuje glavno okno aplikacije.



Slika 25: Glavno okno aplikacije.

4.5 Poizvedbe za pregled statistike

V MySQL in v vseh ostalih sistemih za upravljanje s podatkovnimi bazami (angl. database management system) se podatke izpisuje s pomočjo poizvedb SQL. Za pregled statistike »po materialu« in »po podjetjih« smo v programskem jeziku C# napisali štiri poizvedbe SQL, ki jih bomo podrobno opisali v naslednjih primerih.

Tabela 10 prikazuje primer poizvedbe SQL, napisane v programskem jeziku C#, ki izpiše količino pripeljanega materiala v izbranem časovnem obdobju.

```
@" SELECT m.naziv as 'Material',sum(t.MasaVhod-t.MasaIzhod) as 'Masa [kg]'  
FROM tehtnica.tehtanja t, tehtnica.materiali m  
WHERE t.idMateriala=m.idMateriala AND  
t.casVhod>'"+ dtOd.Value.Date.ToString ("yyyy-MM-dd HH:mm") + "' AND  
t.casIzhod<'"+ dtDo.Value.Date.AddDays(1).ToString("yyyy-MM-dd HH:mm")+ "'  
GROUP BY m.naziv; "
```

Tabela 10: Poizvedba SQL, napisana v programskem jeziku C#, ki izpiše količino pripeljanega materiala v izbranem časovnem obdobju.

Tabela 11 prikazuje primer poizvedbe SQL, napisane v programskem jeziku C#, ki izpiše količino odpeljanega materiala v izbranem časovnem obdobju.

```
@" SELECT m.naziv as 'Material',sum(t.MasaIzhod-t.MasaVhod) as 'Masa [kg]'  
FROM tehtnica.tehtanja t, tehtnica.materiali m  
WHERE t.idMateriala=m.idMateriala AND  
t.casVhod>'"+ dtOd.Value.Date.ToString ("yyyy-MM-dd HH:mm") + "' AND  
t.casIzhod<'"+ dtDo.Value.Date.AddDays(1).ToString("yyyy-MM-dd HH:mm")+ "'  
GROUP BY m.naziv; "
```

Tabela 11: Poizvedba SQL, napisana v programskem jeziku C#, ki izpiše količino odpeljanega materiala v izbranem časovnem obdobju.

Poizvedba SQL, napisana v programskem jeziku C#, je sestavljena iz treh stavkov (SELECT, FROM, WHERE). V našem primeru smo naredili dve poizvedbi, ki izpišeta količino pripeljanega in količino odpeljanega materiala v izbranem časovnem obdobju. S stavkom SELECT določimo attribute, katere želimo izpisati (m.naziv as 'Material', sum(t.MasaIzhod-t.MasaIzhod) as 'Masa [kg]'). S stavkom FROM določimo tabele, katere bomo uporabili (tehtnica.tehtanja t, tehtnica.materiali m). S stavkom WHERE določimo pogoje za izpis. Ker gre za dve povezani tabeli, povežemo primarni ključ prve tabele (m.idMateriala) in tuji ključ druge tabele (t.idMateriala). S stavkom AND določimo še vhodna parametra, in sicer datum vhodnega tehtanja in datum izhodnega tehtanja (t.casVhod>'"+ dtOd.Value.Date.ToString ("yyyy-MM-dd HH:mm"), t.casIzhod<'"+ dtDo.Value.Date.AddDays(1).ToString("yyyy-

MM-dd HH:mm"))). Dobljene podatke uredimo tako, da vrstice v tabeli razdelimo v skupine po materialih. Pri tem uporabimo ukaz GROUP BY (m.naziv).

Tabela 12 prikazuje primer poizvedbe SQL, napisane v programskem jeziku C#, ki za vsa podjetja izpiše količino pripeljanega materiala v izbranem časovnem obdobju.

```
@ " SELECT p.naziv as 'Podjetje',m.naziv as 'Material',
      sum(t.MasaVhod-t.MasaIzhod) as 'Masa [kg]'
FROM tehnica.podjetja p,tehnica.tehtanja t, tehnica.materiali m
WHERE p.idPodjetja=t.idPodjetja AND
      t.idMateriala = m.idMateriala AND
      t.casVhod>" + dtOd.Value.Date.ToString("yyyy-MM-dd HH:mm") + "' AND
      t.casIzhod<" + dtDo.Value.Date.AddDays(1).ToString("yyyy-MM-dd HH:mm")+ "'
GROUP BY p.naziv,m.naziv; "
```

Tabela 12: Poizvedba SQL, napisana v programskem jeziku C#, ki za vsa podjetja izpiše količino pripeljanega materiala v izbranem časovnem obdobju.

Tabela 13 prikazuje primer poizvedbe SQL, napisane v programskem jeziku C#, ki za vsa podjetja izpiše količino odpeljanega materiala v izbranem časovnem obdobju.

```
@ " SELECT p.naziv as 'Podjetje',m.naziv as 'Material',
      sum(t.MasaIzhod-t.MasaVhod) as 'Masa [kg]'
FROM tehnica.podjetja p, tehnica.tehtanja t, tehnica.materiali m
WHERE p.idPodjetja=t.idPodjetja AND
      t.idMateriala = m.idMateriala AND
      t.casVhod>" + dtOd.Value.Date.ToString("yyyy-MM-dd HH:mm") + "' AND
      t.casIzhod<" + dtDo.Value.Date.AddDays(1).ToString("yyyy-MM-dd HH:mm")+ "'
GROUP BY p.naziv,m.naziv; "
```

Tabela 13: Poizvedba SQL, napisana v programskem jeziku C#, ki za vsa podjetja izpiše količino odpeljanega materiala v izbranem časovnem obdobju.

Poizvedba SQL, napisana v programskem jeziku C#, je sestavljena iz treh stavkov (SELECT, FROM, WHERE). V našem primeru smo naredili dve poizvedbi, ki izpišeta količino pripeljanega in količino odpeljanega materiala za vsa podjetja v izbranem časovnem obdobju. S stavkom SELECT določimo, katere attribute želimo izpisati (p.naziv as 'Podjetje', m.naziv as 'Material', sum(t.MasaIzhod-t.MasaIzhod) as 'Masa [kg]'). S stavkom FROM določimo tabele, katere bomo uporabili (tehnica.podjetja p, tehnica.tehtanja t, tehnica.materiali m). S stavkom WHERE določimo pogoje za izpis. Ker gre za tri povezane tabele, povežemo primarni ključ prve tabele (p.idPodjetja) in tuji ključ druge tabele (t.idPodjetja) ter tuji ključ druge tabele (t.idMateriala) in primarni ključ tretje tabele (m.idMateriala) . S stavkom AND določimo še vhodna parametra, in sicer datum vhodnega tehtanja in datum izhodnega tehtanja (t.casVhod>" + dtOd.Value.Date.ToString ("yyyy-MM-dd HH:mm"), t.casIzhod<" + dtDo.Value.Date.AddDays(1).ToString("yyyy-MM-dd HH:mm")). Dobljene podatke uredimo

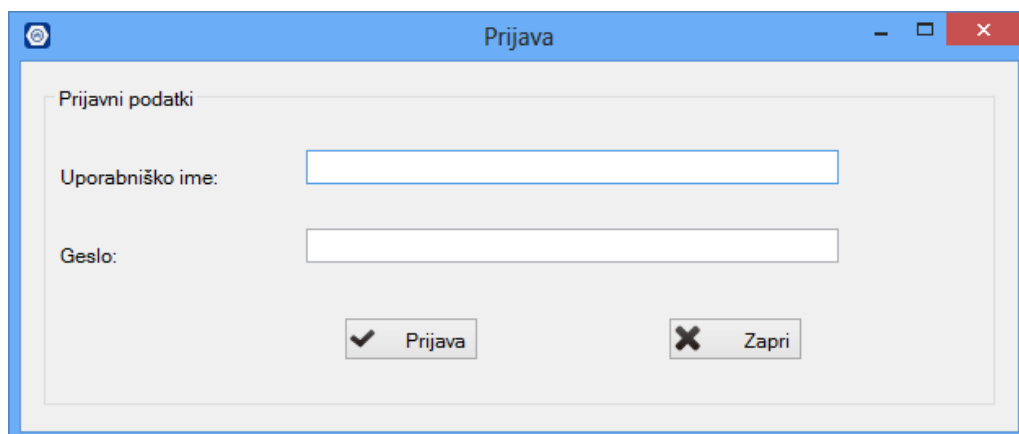
tako, da vrstice v tabeli razdelimo v skupine po materialih in podjetjih. Pri tem uporabimo ukaz GROUP BY (p.naziv, m.naziv).

Poglavje 5 **Predstavitev in uporaba aplikacije kot jo vidi uporabnik**

V tem poglavju bomo predstavili našo aplikacijo in opisali njene funkcionalnosti. Poglavje bo razdeljeno na devet različnih podpoglavij, vsako pa bo opisovalo svojo funkcionalnost.

5.1 Prijava v aplikacijo

Ob zagonu aplikacije se uporabniku odpre prijavno okno (Slika 26), kjer se lahko z vnosom uporabniškega imena in gesla prijavi v aplikacijo. Uporabnik v vnosno polje »Uporabniško ime« in »Geslo« vnese uporabniško ime in geslo, ki mu ga je dodelil administrator. Namesto znakov, ki sestavljajo geslo, se v vnosnem polju »Geslo« zaradi varnostnih razlogov pokažejo znaki »*«. Z gumbom »Prijava« potrdi prijavo v aplikacijo, preklic prijave pa izbere z gumbom »Prekliči«. V primeru napačnega vpisa uporabniškega imena in gesla prijava v aplikacijo ni mogoča. Aplikacija uporabnika obvesti na napačen vnos uporabniškega imena in gesla in mu omogoči ponoven vnos. Ob uspešni prijavi se uporabniku odpre glavno okno aplikacije (Slika 25).



Slika 26: Prijava v aplikacijo.

5.2 Podjetje

Tabela podjetij je namenjena temu, da redne stranke in dobavitelje vnesemo in jih kasneje samo izbiramo, ne da bi pri tem morali vsakič znova vnašati vse podatke podjetja, kot so naziv, naslov, mesto, telefon in e-mail. K vsakemu podjetju lahko v nadaljevanju enostavno dodelimo tudi materiale in vozila. To pomeni, da ko bomo v osnovnem oknu izbrali podjetje, nato pa material, se bodo avtomatično izpisali le materiali, ki pripadajo temu podjetju.

5.2.1 Vnos novega podjetja

V glavnem meniju izberemo meni podjetij in nato v novem oknu pritisnemo gumb »Dodaj«.

Slika 27: Dodajanje podjetja.

Naziv, naslov in mesto podjetja se izpisujejo tudi na tehtalnem listu, medtem ko polje telefon in e-mail podjetja služi le za interne namene (Slika 27). Vnos zaključimo s pritiskom gumba »Dodaj«.

Ko vnesemo podatke podjetja, mu v nadaljevanju dodelimo še materiale in vozila, kot že omenjeno se bodo potem ti materiali in vozila prikazali v iskalnem oknu, ko bomo v glavnem oknu aplikacije izbrali podjetje, nato pa še material in vozilo.

5.2.2 Spreminjanje podatkov obstoječega podjetja

V glavnem meniju izberemo meni podjetja in nato v oknu dvakrat kliknemo podjetje, ki ga želimo spremeniti. Postopek urejanja je enak vnosu, le da tukaj že imamo vnesene podatke, ki jih samo spremenimo. Spremembe potrdimo z gumbom »Shrani«.

5.2.3 Brisanje podjetja

V glavnem meniju izberemo meni podjetja in nato v oknu izberemo podjetje, ki ga želimo odstraniti, ter pritisnemo tipko »Delete« ali gumb »Odstrani«. V pogovornem oknu potrdimo brisanje in podjetje bo odstranjeno.

5.2.4 Izvoz v CSV in tiskanje poročil

Vsako podjetje ali posamezna podjetja lahko izvozimo v datoteko CSV. To storimo tako, da označimo podjetja, katerih podatke želimo izvoziti, ter pritisnemo desni gumb miške in v kontekstnem meniju izberemo »Izvoz v CSV«. Nato aplikacija ponudi pogovorno okno, v katerem izberemo lokacijo, kam se naj izvoz izvrši, in vnesemo tudi ime datoteke.

Postopek tiskanja poročil je enak postopku izvoza podatkov v CSV, vendar tukaj, ko izberemo želena podjetja, le pritisnemo gumb »Natisni« ali »Predogled tiskanja«, v kolikor želimo poročilo pred tiskanjem videti.

5.3 Material

Tabela materialov je namenjena temu, da vse materiale, ki se redno tehtajo, vnesemo v podatkovno bazo in jih kasneje le izbiramo ter dodeljujemo posameznim podjetjem.

5.3.1 Vnos novega materiala

V glavnem meniju izberemo meni materialov in nato v novem oknu pritisnemo gumb »Dodaj«.

Slika 28: Dodajanje materiala.

Naziv in šifra materiala se izpisujeta tudi na tehtalnem listu, med-tem ko polje opis materiala služi le za interne namene (Slika 28). Vnos zaključimo s pritiskom gumba »Dodaj«.

5.3.2 Spreminjanje podatkov obstoječega materiala

V glavnem meniju izberemo meni materialov in nato v oknu dvakrat kliknemo material, ki ga želimo spremeniti. Postopek spreminjanja je enak vnosu, le da tukaj že imamo vnesene podatke, ki jih samo spremenimo. Spremembo potrdimo z gumbom »Shrani«.

5.3.3 Brisanje materiala

V glavnem meniju izberemo meni materialov in nato v oknu izberemo material, ki ga želimo odstraniti, ter pritisnemo tipko »Delete« ali gumb »Odstrani«. V pogovornem oknu potrdimo brisanje in material bo odstranjen.

5.3.4 Izvoz v CSV in tiskanje poročil

Vsak material ali posamezne materiale lahko izvozimo v datoteko CSV. To storimo tako, da označimo materiale, katerih podatke želimo izvoziti, ter pritisnemo desni gumb miške in v kontekstnem meniju izberemo »Izvoz v CSV«. Nato aplikacija ponudi pogovorno okno, v katerem izberemo lokacijo, kam se naj izvoz izvrši, in vnesemo tudi ime datoteke.

Postopek tiskanja poročil je enak postopku izvoza podatkov v CSV, vendar tukaj, ko izberemo želene materiale, le pritisnemo gumb »Natisni« ali »Predogled tiskanja«, v kolikor želimo poročilo pred tiskanjem videti.

5.4 Vozila

Tabela vozil je namenjena temu, da vsa vozila, ki redno vozijo v podjetje, vnesemo v podatkovno bazo in jih kasneje le izbiramo ter dodeljujemo posameznim podjetjem.

5.4.1 Vnos novega vozila

V glavnem meniju izberemo meni vozil in nato v novem oknu pritisnemo gumb »Dodaj«.

Slika 29: Dodajanje vozila.

Vsebina polja registrska številka vozila ter voznik vozila se izpisuje tudi na tehtalnem listu, med-tem ko polje opis in največja dovoljena masa služi internim opombam (Slika 29). Vnos potrdimo s pritiskom gumba »Dodaj«.

5.4.2 Spreminjanje podatkov obstoječega vozila

V glavnem meniju izberemo meni vozil in nato v oknu dvakrat kliknemo vozilo, ki ga želimo spremeniti. Postopek spreminjanja je enak vnosu, le da tukaj že imamo vnesene podatke, ki jih samo spremenimo. Spremembo potrdimo z gumbom »Shrani«.

5.4.3 Brisanje vozila

V glavnem meniju izberemo meni vozil in nato v oknu izberemo vozilo, ki ga želimo odstraniti, ter pritisnemo tipko »Delete« ali gumb »Odstrani«. V pogovornem oknu potrdimo brisanje in vozilo bo odstranjeno.

5.4.4 Izvoz v CSV in tiskanje poročil

Vsako vozilo ali posamezna vozila lahko izvozimo v datoteko CSV. To storimo tako, da označimo vozila, katerih podatke želimo izvoziti, ter pritisnemo desni gumb miške in v kontekstnem meniju izberemo »Izvoz v CSV«. Nato aplikacija ponudi pogovorno okno, v katerem izberemo lokacijo, kam se naj izvoz izvrši, in vnesemo tudi ime datoteke.

Postopek tiskanja poročil je enak postopku izvoza podatkov v CSV, vendar tukaj, ko izberemo zelena vozila, le pritisnemo gumb »Natisni« ali »Predogled tiskanja«, v kolikor želimo poročilo pred tiskanjem videti.

5.5 Tehtanja

Tabela tehtanj služi shranjevanju vseh opravljenih tehtanj (zgodovina tehtanj). V tej tabeli so shranjena le zaključena tehtanja, tj. brez odprtih tehtanj (Slika 30).

Kriteriji filtriranja tehtanj.

Zadetki glede na iskalne kriterije.

Kontekstni meni (desni klik).

| ID | Podjetje | Material | Šifra m. | RegSt | Voznik | Datum/Ura VT | Masa VT [kg] | Uporabnik VT | Datum/Ura IT | Masa IT [kg] | Uporabnik IT | Razlika VI [kg] |
|-----|--------------------|------------|-----------|-----------|-------------|----------------|--------------|--------------|----------------|--------------|--------------|-----------------|
| 105 | YO&CO | Kemikalije | DC-566788 | LJ-85-78 | Janez Novak | 8.7.2014 16:45 | 0,70 | m r | 8.7.2014 16:45 | 0,02 | m r | 0,68 |
| 104 | AHATServis, d.o.o. | Les | BC-34567 | KR-111-KK | Janez Novak | 6.7.2014 11:13 | 1000,00 | m r | 8.7.2014 16:43 | 0,70 | m r | 999,30 |
| 99 | YO&CO | Kemikalije | DC-566788 | LJ-85-78 | Janez Novak | 6.7.2014 10:42 | 0,69 | m r | 6.7.2014 10:42 | 0,01 | m r | 0,68 |
| 97 | AHATServis, d.o.o. | Les | BC-34567 | KR-111-KK | Janez Novak | 5.7.2014 19:48 | 0,69 | m r | 5.7.2014 19:48 | 0,02 | m r | 0,67 |
| 96 | YO&CO | Kemikalije | DC-566788 | LJ-85-78 | Janez Novak | 5.7.2014 19:47 | 0,16 | m r | 5.7.2014 19:48 | 0,01 | m r | 0,15 |
| 95 | AHATServis, d.o.o. | Pesek | AC-123698 | MB-369-LK | Janez Novak | 5.7.2014 17:25 | 0,17 | m r | 5.7.2014 17:25 | 0,17 | m r | 0,58 |

*VT - vhodno tehtanje | IT - izhodno tehtanje

Število zadetkov: 6

Slika 30: Pregled opravljenih oziroma zaključenih tehtanj.

Iskalni kriterij je lahko naziv podjetja ali naziv materiala ali registrska številka vozila. Iskanje lahko omejimo tudi datumsko, in sicer tako, da v poljih datum od in datum do vnesemo oziroma izberemo datum.

V kontekstnem meniju lahko izberemo:

- predogled tiskanja tehtalnega lista;
- tiskanje tehtalnega lista;
- ponovno odpremo tehtanje (v tem primeru se podatki izhodnega tehtanja izbrišejo izbrano tehtanje pa se ponovno pojavi na seznamu odprtih tehtanj) ali
- izvozimo podatke izbranih tehtanj v datoteko CSV za nadaljnjo obdelavo, npr. v Excelu.

Tehtanja je možno tudi odstraniti, kar storimo tako, da tehtanje ali tehtanja označimo in pritisnemo tipko »Delete«.

V kolikor se tehtanje odstrani, se ustrezno posodobijo vse statistike, ki se nanašajo na podjetje, in material, ki je bil izbran pri tem tehtanju. Izbrisana tehtanja še vedno ostanejo shranjena v podatkovni bazi, vendar, kot že rečeno, se ne upoštevajo več pri statistikah. Z brisanjem tehtanj se izbriše tudi št. tehtalnega lista, zato v primeru, da je bilo tehtanje odstranjeno, ta zaporedna številka manjka.

5.6 Izvajanje tehtanj

Praviloma izvajamo vhodno/izhodna tehtanja. To pomeni, da vozilo ob prihodu v podjetje stehtamo, kar predstavlja vhodno tehtanje. Pred izhodom iz podjetja vozilo ponovno stehtamo, kar predstavlja izhodno tehtanje. Ko je opravljeno vhodno in izhodno tehtanje, je tehtanje zaključeno. Na tehtalnem listku se potem izpišejo vsi podatki tehtanja, pri čemer se avtomatsko določi, ali je vozilo pripeljalo material (masa vhodnega tehtanja je večja od mase izhodnega tehtanja) ali odpeljalo material (masa vhodnega tehtanja je manjša od mase izhodnega tehtanja).

5.6.1 Vnos (izbor) podatkov

Preden lahko pričnemo s tehtanjem, moramo najprej v osnovni maski programa izbrati podjetje, vozilo in material (Slika 31). Dokler teh podatkov ni, tehtanje ni možno. V kolikor vozilo, ki je pripeljalo material, pripada podjetju, ki je vneseno v bazi podjetij, le-to enostavno izberemo.

The screenshot shows the ATS application window. At the top, there's a title bar with 'ATS' and standard window controls. Below it is a toolbar with icons for home, back, forward, and search. The main area is divided into sections. The top section, titled 'Masa', displays a large black box with a blue '=' sign and '0,00 kg' in blue. Below this are two buttons: 'TARIRAJ' and 'NIČLIRAJ'. To the right of these buttons is the timestamp '11.7.2014 18:29:34'. Below the buttons are three selection fields: 'Podjetje:', 'Vozilo:', and 'Material:'. Each field has a blue 'X' icon and a dropdown menu. Three blue callout boxes point to these fields with the text 'Izbor podjetja.', 'Izbor vozila.', and 'Izbor materiala.' respectively. Below the selection fields is a section titled 'Odprta tehtanja' which contains a table with the following columns: 'Datum/Ura', 'Podjetje', 'Material', 'Šifra m.', 'Reg. št.', 'Voznik', and 'Masa [kg]'. The table body is currently empty. At the bottom of the window, there are two buttons: 'Vhodno tehtanje' (with a truck icon) and 'Izhodno tehtanje' (with a factory icon). In the center of the bottom section is a text box labeled 'Obvestila' containing the text 'Zagon ATS aplikacije - 11.7.2014 18:29:32'. The bottom status bar shows 'Uporabnik: m r'.

Masa

= 0,00 kg

TARIRAJ NIČLIRAJ 11.7.2014 18:29:34

Podjetje: X ... Izbor podjetja.

Vozilo: X ... Izbor vozila.

Material: X ... Izbor materiala.

Odprta tehtanja

| Datum/Ura | Podjetje | Material | Šifra m. | Reg. št. | Voznik | Masa [kg] |
|-----------|----------|----------|----------|----------|--------|-----------|
|-----------|----------|----------|----------|----------|--------|-----------|

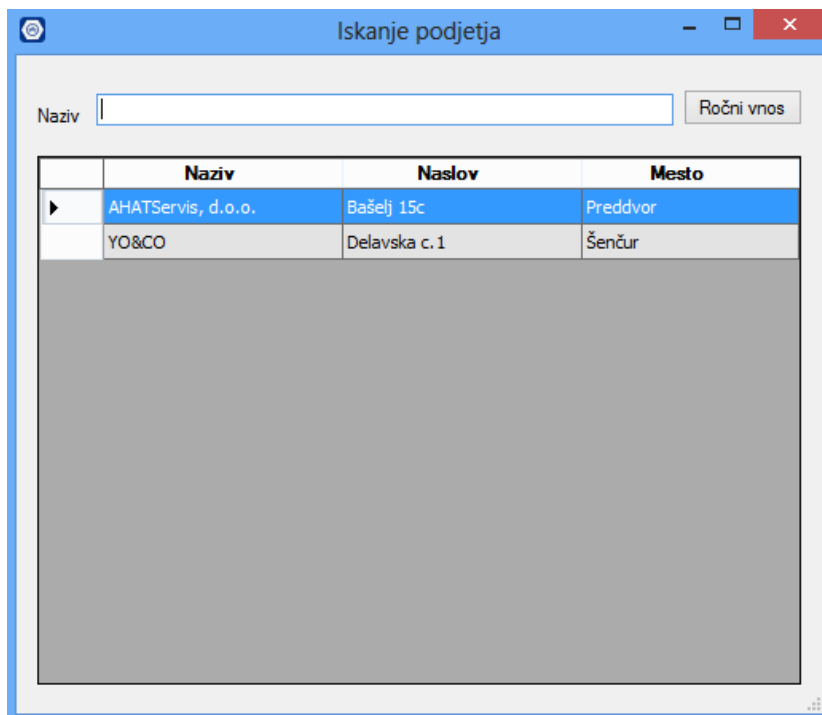
Vhodno tehtanje Obvestila Izhodno tehtanje

Obvestila
Zagon ATS aplikacije - 11.7.2014 18:29:32

Uporabnik: m r

Slika 31: Vnos podatkov o podjetju, vozilu in materialu.

Ko kliknemo na gumb »...« za izbor podjetja, se odpre okno »Iskanje podjetja«.



Slika 32: Iskanje podjetja.

V oknu »Iskanje podjetja« (Slika 32) lahko podjetje z dvoklikom izberemo iz seznama ali pa ga poiščemo na podlagi naziva, ki ga vnesemo v vnosno polje naziv.

Kadar izberemo podjetje iz baze, bomo pri izboru materiala/vozila videli le tiste materiale/vozila, ki so dodeljeni temu podjetju.

Ko izberemo podjetje, lahko nato izberemo material in vozilo. Postopek je enak izboru podjetja.

5.6.2 **Brisanje oziroma odznačitev vnesenih podatkov**

Vnesene podatke pobrišemo tako, da pritisnemo na »X« pred gumbom ». .«. V pogovornem oknu pa potrdimo brisanje vnesenih podatkov.

5.6.3 **Izvedba tehtanja**

Najprej izvedemo vnos podatkov (Slika 33), ki je opisan v razdelku 5.6.1. Ko opravimo vnos podatkov, se na semaforju prižge zelena luč, kar pomeni, da vozilo lahko zapelje na tehtnico. Ko vozilo zapelje na tehtnico, se prižge rdeča luč, ki pomeni, da je tehtanje v teku. Sedaj pritisnemo na gumb »Vhodno tehtanje« (Slika 33). V tem trenutku čakamo na tri zaporedne pozitivne meritve mase v intervalnem zamiku ene sekunde. Ko program zabeleži tri identične meritve, zabeleži trenutno maso in ostale podatke, ki se nanašajo na meritev, v tabelo odprtih

tehtanj. Ko je postopek zaključen, se na semaforju prižge zelena luč, kar pomeni, da lahko vozilo zapusti tehtnico. Ko vozilo zapusti tehtnico, se na semaforju ponovno prižge rdeča luč.

Ko se vozilo vrne na izhodno tehtanje, moramo ustrezno tehtanje izbrati na seznamu odprtih tehtanj (Slika 33). Z dvoklikom potrdimo izbor in sedaj se vsi vneseni podatki prenesejo nazaj v izbor. Na semaforju se prižge zelena luč, kar pomeni, da vozilo lahko zapelje na tehtnico. Ko vozilo zapelje na tehtnico, se prižge rdeča luč, ki pomeni, da je tehtanje v teku. Sedaj pritisnemo na gumb »Izhodno tehtanje« (Slika 33). V tem trenutku čakamo na tri zaporedne pozitivne meritve mase v intervalnem zamiku ene sekunde. Ko program zabeleži tri identične meritve, zabeleži trenutno maso in ostale podatke, ki se nanašajo na meritev, v tabelo zaključenih tehtanj. Ko je postopek zaključen, se na semaforju prižge zelena luč, kar pomeni, da lahko vozilo zapusti tehtnico. Ko vozilo zapusti tehtnico, se na semaforju ponovno prižge rdeča luč.



Slika 33: Izvedba tehtanja.

5.6.4 Tehtalni list

Ko je opravljeno vhodno in izhodno tehtanje, je tehtanje zaključeno in se lahko izda tehtalni list (Slika 35). Tehtalni list vsebuje naslednje podatke:

- naziv in naslov podjetja, ki izda tehtalni list;
- številko tehtalnega lista;
- naziv in naslov podjetja, ki tehtalni list prejme;
- naziv in šifra materiala;
- registrska številka vozila;
- ime in priimek voznika;
- ime in priimek operatorja tehtnice;
- datum in ura vhodnega tehtanja;
- masa vhodnega tehtanja v kilogramih;
- datum in ura izhodnega tehtanja;
- masa izhodnega tehtanja v kilogramih;
- bruto masa v kilogramih (tj. skupna masa);
- tara v kilogramih (tj. razlika med bruto maso in neto maso);
- neto masa v kilogramih (tj. razlika med bruto maso in taro);
- podpis operaterja tehtnice;
- podpis voznika vozila.

Tehtalni list natisnemo tako, da v tabeli zaključenih tehtanj izberemo tehtanje, kliknemo desni miškin gumb in v kontekstnem meniju izberemo »Tiskanje tehtalnega lista« ali »Predogled tiskanja«, v kolikor želimo tehtalni list pred tiskanjem videti (Slika 34).

Tehtanja

| | ID | Podjetje | Material | Šifra m. | RegSt | Voznik |
|---|-----|--------------------|----------|-----------|----------|-------------|
| ▶ | 109 | AHATServis, d.o.o. | Les | DC-566788 | LJ-85-78 | Janez Novak |

Predogled tiskanja
 Natisni tehtalni list
 Ponovno odpri vhodno tehtanje
 Izvozi tabelo v CSV

Slika 34: Tiskanje tehtalnega lista.

AHATServis, d.o.o.

Bašelj 15/c, 4205 Preddvor

T: 04 / 25 55 310 | F: 04 / 25 55 311

E: prodaja@ahatservis.si | W: www.ahatservis.si

TEHTALNI LIST: 114/2014

Podjetje: YO&CO, Delavska c.1, Šenčur

Material: Kemikalije (šifra: DC-566788)

Reg. št. vozila: LJ-85-78

Voznik: Janez Novak

Operator tehtnice: Mitja Roblek

Podatki vhodnega
tehtanja.

Vhodno tehtanje - 29.7.2014 15:12:08

Vhodna masa: 60000,00 kg

Izhodno tehtanje - 29.7.2014 15:12:10

Izhodna masa: 10000,00 kg

Bruto: 60000,00 kg

Tara: 10000,00 kg

Neto razlika (pripeljano): 50000,00 kg

Podatki izhodnega
tehtanja.

Podpis operaterja: _____

Podpis voznika: _____

Vse pravice pridržane!

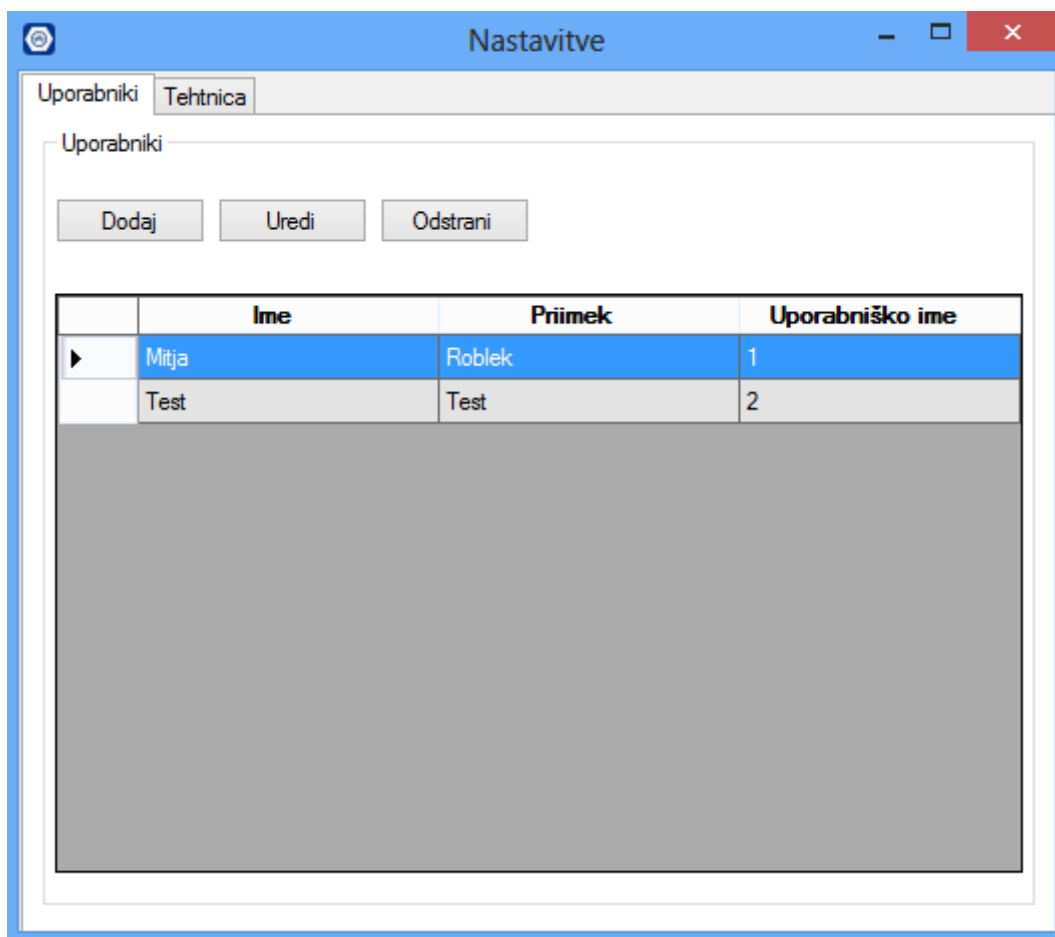
Slika 35: Primer tehtalnega lista za vhodno/izhodno tehtanje.

5.7 Nastavitve

V glavnem meniju izberemo meni »Nastavitve« (Slika 36). V »Nastavitvah« imamo dva zavihka »Uporabniki« in »Tehcnica«, kjer lahko nastavljamo:

- parametre za komunikacijo s tehtnico preko serijskega vmesnika;
- dodajamo, spreminjamo in odstranjujemo končne uporabnike programa (vsakemu uporabniku je možno dodeliti vlogo oziroma tip uporabnika).

Do modula nastavitvev lahko dostopajo le uporabniki, ki imajo vlogo administratorja. Ti uporabniki lahko vzpostavijo komunikacijo s tehtnico in opravljajo s končnimi uporabniki.

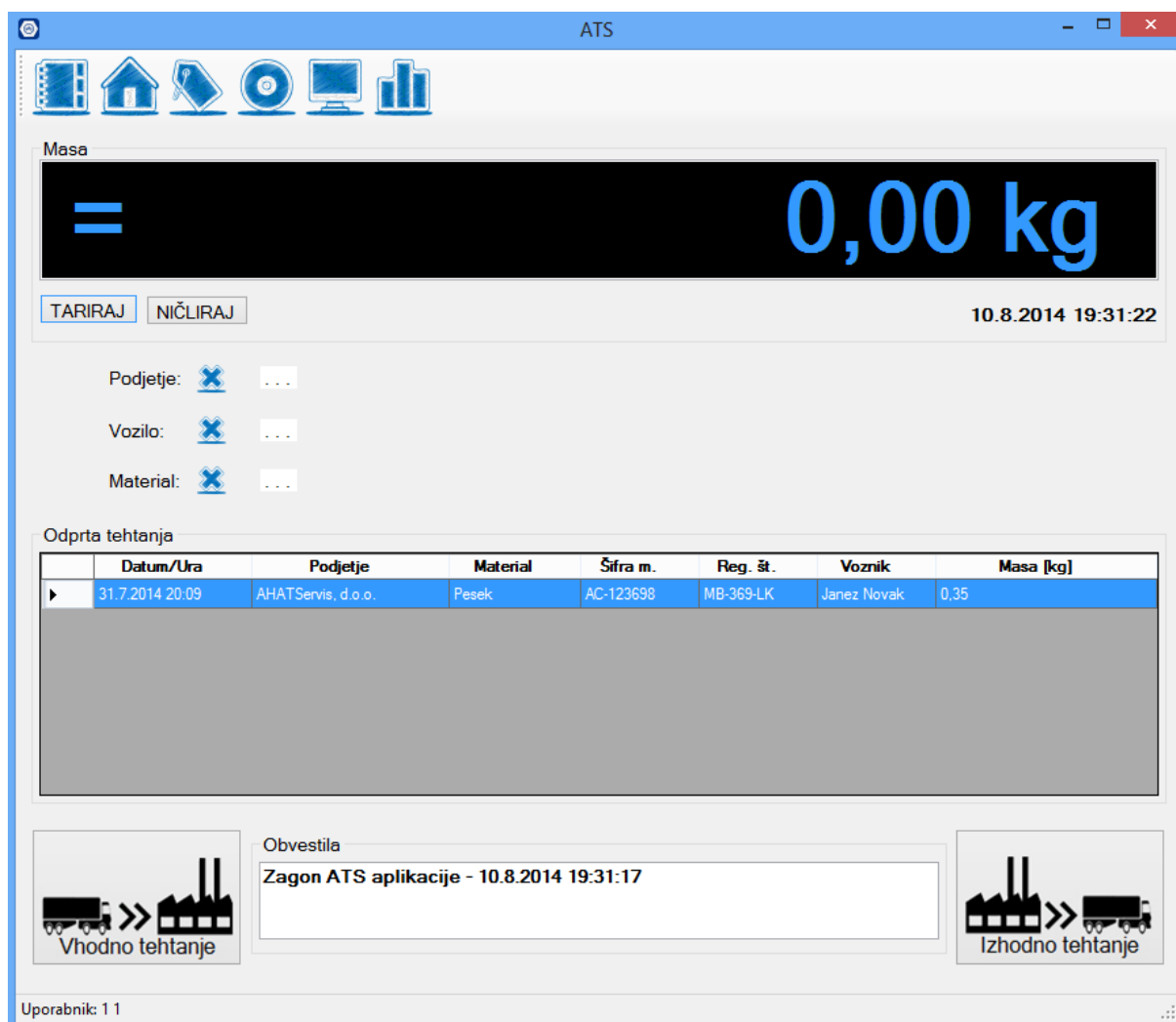


Slika 36: Nastavitve.

5.7.1 Uporabniki

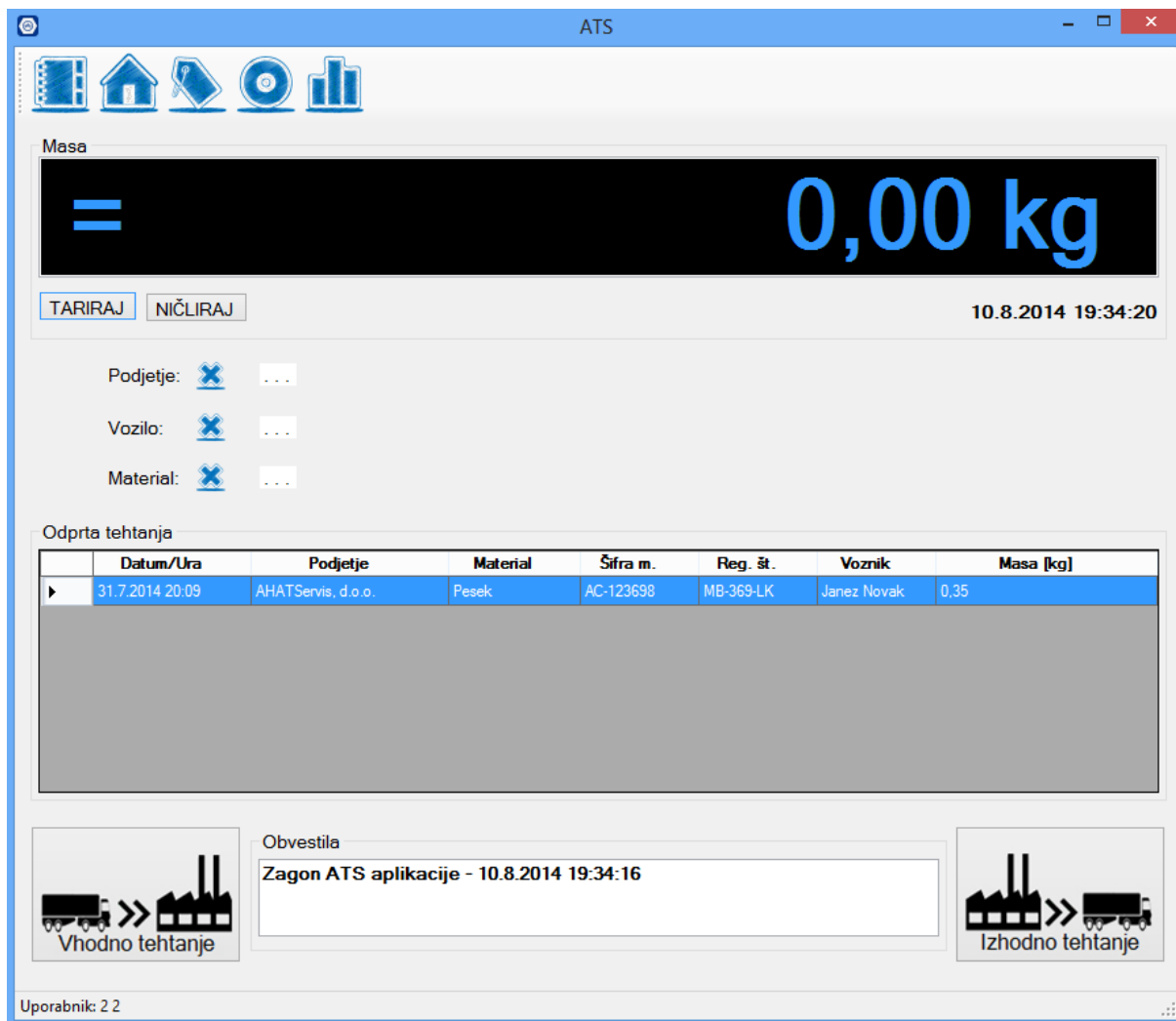
Program ob zagonu zahteva uporabniško ime in geslo. Z uporabniki identificiramo osebo, ki izvaja tehtanje. Zavihek »Uporabniki« je namenjen kreiranju, spreminjanju in brisanju končnih uporabnikov (»end customer«). Vsakemu uporabniku moramo določiti ime, priimek, uporabniško ime, geslo in pa vlogo oziroma tip uporabnika. Uporabniki so lahko tipa »administrator« ali »end customer«. Končni uporabniki (»end customer«) imajo omejene pravice.

Uporabniki tipa administrator imajo dostop do vseh menijev (Slika 37) v aplikaciji, kreirajo lahko nove uporabnike tipa končni uporabnik (»end customer«) in spreminjajo komunikacijske parametre za vzpostavljanje komunikacije s tehtnico.



Slika 37: Aplikacija ATS (logiran je administrator).

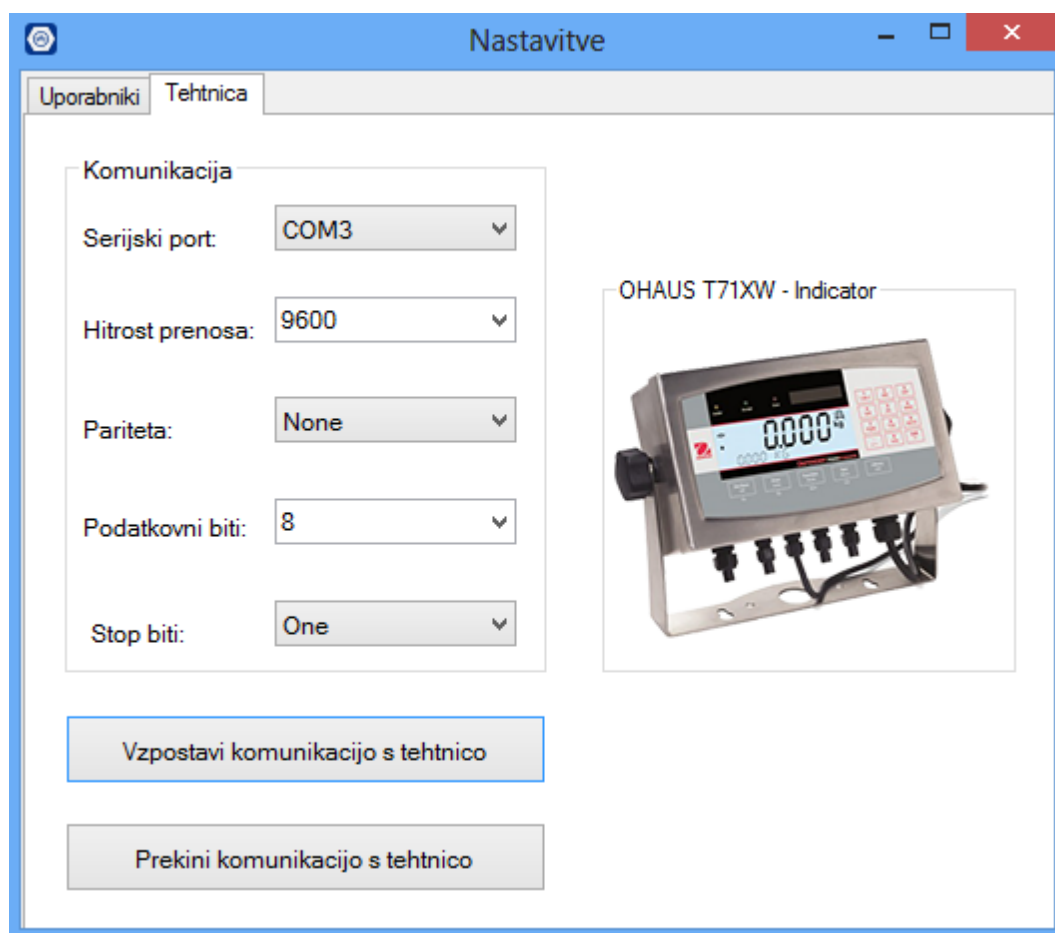
Uporabniki tipa končni uporabnik (»end customer«) pa nimajo dostopa do menija »Nastavitev« (Slika 38), tako da ne morejo kreirati novih uporabnikov in spreminjati komunikacijskih parametrov za vzpostavljanje komunikacije s tehtnico.



Slika 38: Aplikacija ATS (logiran je končni uporabnik).

5.7.2 Tehnica

Zavihek »Tehnica« (Slika 39) je namenjen nastavitvam komunikacijskih parametrov za vzpostavljjanje komunikacije s tehcnico preko serijskega vmesnika. Parametri morajo biti nastavljeni enako kot so nastavljeni na tehcnici. Komunikacijo vzpostavimo s pritiskom gumba »Vzpostavi komunikacijo s tehcnico«.



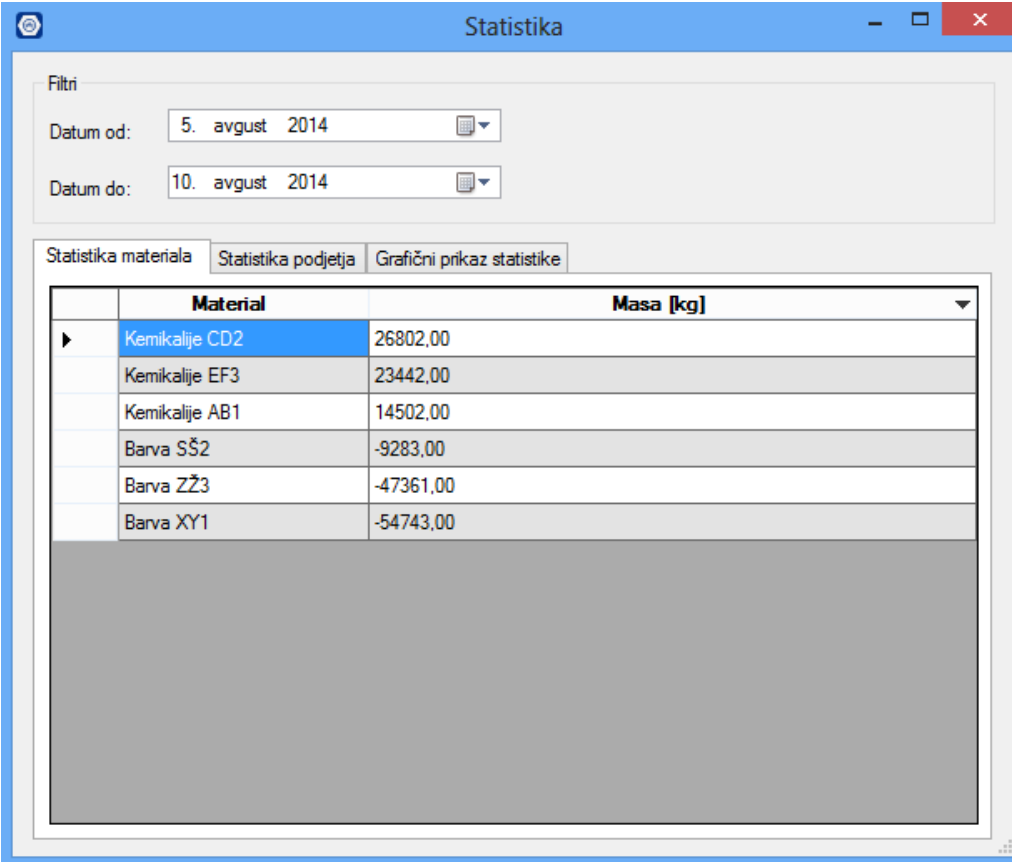
Slika 39: Nastavitve tehcnice.

5.8 Statistika

Aplikacija nam omogoča pregled statistike po dnevih v izbranem časovnem obdobju, in sicer lahko spremljamo količino pripeljanega in odpeljanega materiala in količino pripeljanega in odpeljanega materiala po podjetjih. Poizvedbe SQL, napisane v programskem jeziku C#, ki nam omogočajo pregled statistike, se nahajajo v razdelku 4.5.

5.8.1 Po materialih

Statistika materiala nam omogoča pregled količine pripeljanega in odpeljanega materiala v izbranem časovnem obdobju. Statistika materiala se vodi samodejno, takoj ko se zabeleži novo vhodno in izhodno tehtanje. Podatke o količini materiala prikazujemo tabelarično in grafično. Podjetje na ta način lahko spremlja zalogo materiala. V kolikor se tehtanje odstrani, se ustrezno posodobi statistika, ki se nanaša na material. Slika 40 prikazuje količino pripeljanega materiala (pozitivna vrednost) in količino odpeljanega materiala (negativna vrednost) v obdobju od 5.8.2014 do 10.8.2014.

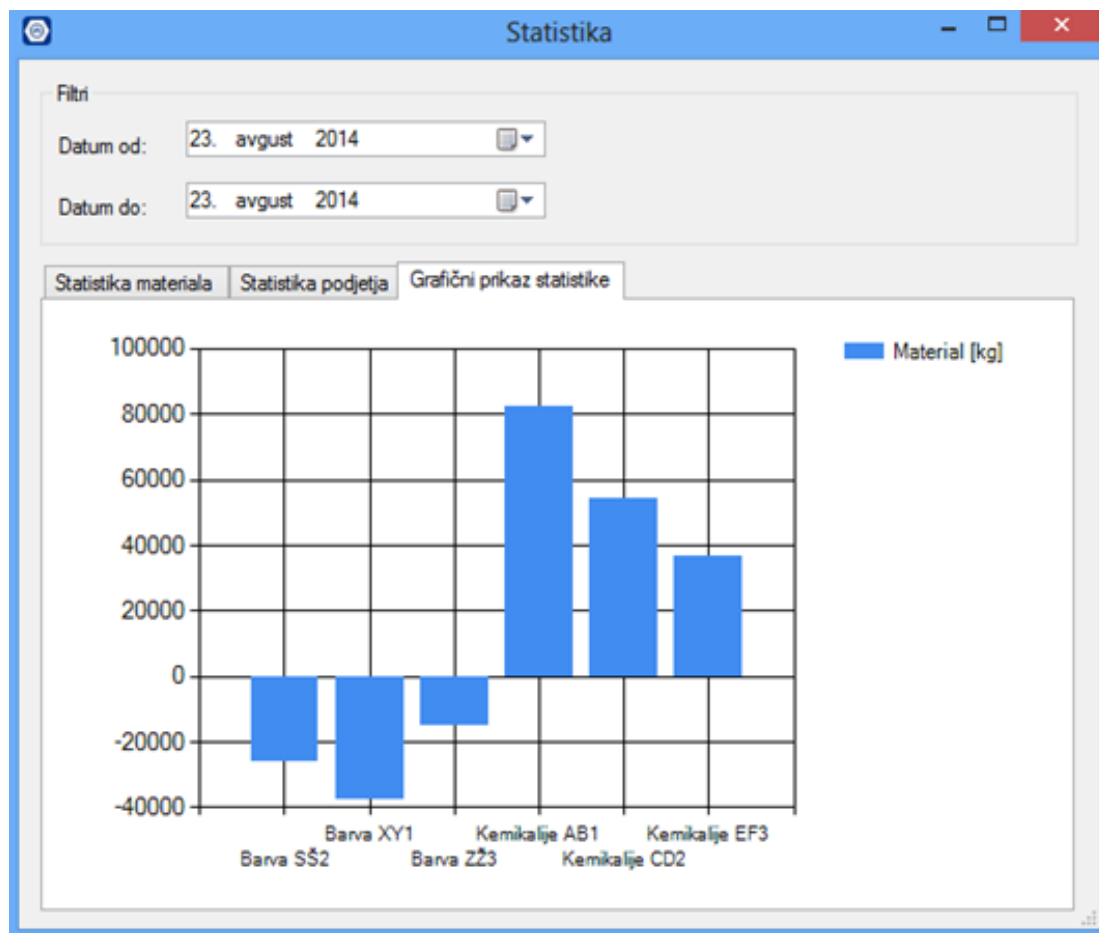


The screenshot shows a window titled "Statistika" with a filter section at the top. The filter section has two date pickers: "Datum od:" set to "5. avgust 2014" and "Datum do:" set to "10. avgust 2014". Below the filter section are three tabs: "Statistika materiala" (selected), "Statistika podjetja", and "Grafični prikaz statistike". The main content area displays a table with two columns: "Material" and "Masa [kg]". The table contains six rows of data, with the first row highlighted in blue.

| Material | Masa [kg] |
|----------------|-----------|
| Kemikalije CD2 | 26802,00 |
| Kemikalije EF3 | 23442,00 |
| Kemikalije AB1 | 14502,00 |
| Barva SŠ2 | -9283,00 |
| Barva ZŽ3 | -47361,00 |
| Barva XY1 | -54743,00 |

Slika 40: Tabelarni prikaz količine pripeljanega materiala (pozitivna vrednost) in količine odpeljanega materiala (negativna vrednost) v obdobju od 5.8.2014 do 10.8.2014.

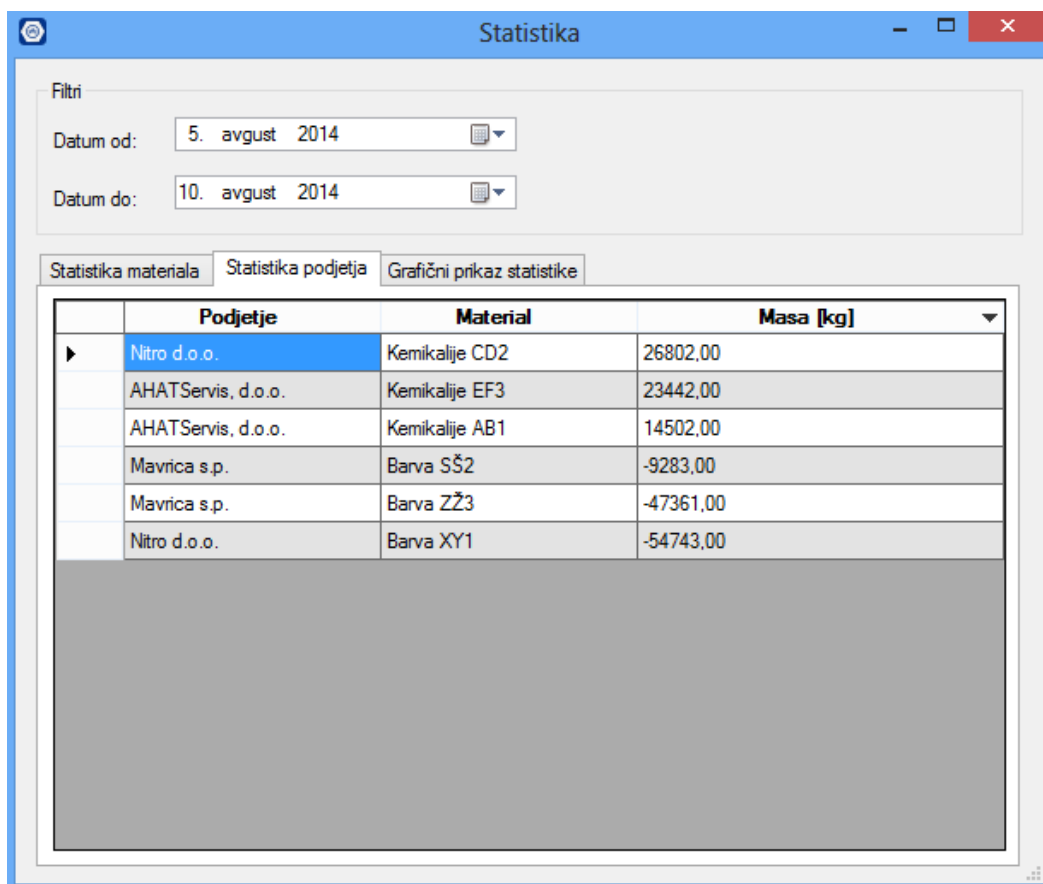
Slika 41 prikazuje količino pripeljanega materiala (pozitivna vrednost) in količino odpeljanega materiala (negativna vrednost) na dan 23.8.2014.



Slika 41: Grafični prikaz količine pripeljanega materiala (pozitivna vrednost) in količine odpeljanega materiala (negativna vrednost) na dan 23.8.2014.

5.8.2 Po podjetjih

Statistika podjetja nam omogoča pregled količine pripeljanega in odpeljanega materiala v izbranem časovnem obdobju za vsa podjetja. Statistika podjetij se vodi samodejno, takoj ko se zabeleži novo vhodno in izhodno tehtanje. Podatke prikazujemo tabelarično. V kolikor se tehtanje odstrani, se ustrezno posodobi statistika, ki se nanaša na podjetje in material. Slika 42 prikazuje količino pripeljanega materiala (pozitivna vrednost) in količino odpeljanega materiala (negativna vrednost) za vsa podjetja v obdobju od 5.8.2014 do 10.8.2014.



Statistika

Filtri

Datum od: 5. avgust 2014

Datum do: 10. avgust 2014

Statistika materiala Statistika podjetja Grafični prikaz statistike

| | Podjetje | Material | Masa [kg] |
|---|--------------------|----------------|-----------|
| ▶ | Nitro d.o.o. | Kemikalije CD2 | 26802,00 |
| | AHATServis, d.o.o. | Kemikalije EF3 | 23442,00 |
| | AHATServis, d.o.o. | Kemikalije AB1 | 14502,00 |
| | Mavnica s.p. | Barva SŠ2 | -9283,00 |
| | Mavnica s.p. | Barva ZŽ3 | -47361,00 |
| | Nitro d.o.o. | Barva XY1 | -54743,00 |

Slika 42: Tabelarni prikaz količine pripeljanega materiala (pozitivna vrednost) in količine odpeljanega materiala (negativna vrednost) za vsa podjetja v obdobju od 5.8.2014 do 10.8.2014.

5.9 Grafična ponazoritev izvajanja tehtanja za voznika

Slika 43 prikazuje grafično ponazoritev izvajanja tehtanja za voznika. Luči so krmiljene neodvisno od računalnika. Krmili jih operater tehtnice.

1 – tehtnica v mirovanju (čakanje na vozilo)



2 – vozilo pripelje do tehtnice in se ustavi pred rdečo lučjo oziroma STOP znakom



3 – zelena luč (voznik je dobil dovoljenje, da zapelje na tehtnico)



4 – vozilo je na tehtnici (čakanje stabilne mase oziroma da se vozilo ustavi)



5 – ko se vozilo ustavi se prične izvajati tehtanje



6 – ko se tehtanje izvede se prižge zelena luč (voznik lahko zapelje iz tehtnice)



7 – vozilo zapusti tehtnico



8 – tehtanje končano



Slika 43: Grafična ponazoritev izvajanja tehtanja za voznika

Poglavje 6 **Sklepne ugotovitve**

V diplomskem delu je bila izdelana namizna aplikacija, ki omogoča beleženje mase vhodnih in izhodnih tehtanj tovornih vozil. Aplikacija je namenjena podjetjem, ki uporabljajo cestno tehtnico, s pomočjo katere preverjajo tako prejete kot tudi izdane količine gotovega blaga, polproizvodov, proizvodov in surovin, ki jih podjetja potrebujejo za svojo dejavnost.

Pri pregledu obstoječih aplikacij smo ugotovili, da imajo le-te nepregledne uporabniške vmesnike, vsebujejo odvečne funkcionalnosti, vsebujejo nepotrebne gumbе, uporabniški vmesniki niso prevedeni v slovenski jezik in so sorazmerno drage (cena ena licence se giblje med 300,00 € in 500,00 €, odvisno od proizvajalca). Z izdelavo lastne aplikacije smo te pomanjkljivosti oziroma slabosti odpravili, ker pa smo aplikacijo razvili sami, jo lahko poljubno prilagajamo potrebam stranke.

Izdelana aplikacija je univerzalna, kar pomeni, da ob menjavi tehtnice ne bo treba menjati aplikacije, temveč samo spremenimo nastavitve protokola tehtnice ter delamo naprej z isto programsko opremo. S tem se izognemo izgubi tehtanj ali migraciji podatkov in ponovnemu uvajanju operaterjev za delo z novo aplikacijo.

Glavna cilja diplomskega dela sta bila pregled protokola, po katerem tehtnica komunicira z računalnikom, in izdelava komunikacijskega protokola po standardu RS-232, ki smo ga implementirali v aplikacijo. S protokolom smo definirali, na kakšen način naj aplikacija sprejema podatke iz tehtnice in iz njih izlušči ustrezne podatke o masi. Aplikacija je izdelana v ogrodju .NET in za hranjenje podatkov uporablja podatkovno bazo MySQL.

Aplikacijo v podjetju skupaj s sodelavci še testiramo, tako bo kmalu na voljo končnim uporabnikom. Po prvih odzivih sodelavcev je videti, da bodo končni uporabniki z našo aplikacijo zadovoljni, saj jim le-ta omogoča hitro in enostavno operiranje s tehtanji tovornih vozil. V podjetju smo se odločili, da bomo strankam aplikacijo ob nakupu cestne tehtnice ponudili zastonj in na ta način povečali svojo konkurenčno prednost.

Z nadaljnjim razvojem bi lahko aplikacijo izboljšali. Aplikacijo se lahko izpopolni na naslednji način:

- tehtanje z daljinskim upravljanjem: daljinci omogočajo izvajanje tehtanja brez operaterja tehtnice. Uporaba daljincev bi bila izjemno priročna, saj pri tehtanju ni potreben operater in prav tako je lahko tehtnica na oddaljeni lokaciji, kjer ni tehtalne hiške ali računalnika, vsi podatki tehtanja pa se vseeno hranijo in tiskajo iz računalnika;
- spletni dostop do aplikacije: spletni dostop do aplikacije bi omogočil pregled zaključenih in odprtih tehtanj z več lokacij in z več različnimi uporabniki. Uporabnik bi imel popolnoma enake možnosti dela kot s primarno aplikacijo, vendar preko spletnega dostopa ne bi bilo možno opravljati tehtanj ter vnašati, spreminjati ali brisati obstoječih tehtanj.

Literatura in viri

- [1] (2014) *Dini Argeo*. Dostopno na:
<http://www.diniargeo.com/lst/scales/vehicle-weighing/software-for-weighbridge.aspx>
- [2] (2014) *Leon Engineering*. Dostopno na:
<http://www.leon-engineering.com/en/weighing-equipment/weighbridge-systems-and-software.html>
- [3] (2014) *Ohaus*. Dostopno na:
<http://europe.ohaus.com/europe/en/home/markets/industrial-world.aspx>
- [4] (2014) *Tehcnica RANGER 3000*. Dostopno na:
<http://dmx.ohaus.com/WorkArea/showcontent.aspx?id=30234>
- [5] (2014) *Digitalni tehtalni indikator T71XW*. Dostopno na:
<http://dmx.ohaus.com/WorkArea/showcontent.aspx?id=18284>
- [6] (2014) *Wikipedia: Microsoft Visual Studio*. Dostopno na:
http://en.wikipedia.org/wiki/Microsoft_Visual_Studio
- [7] (2014) *Wikipedia: MySQL Workbench*. Dostopno na:
http://en.wikipedia.org/wiki/MySQL_Workbench
- [8] (2014) *MySQL Workbench*. Dostopno na:
<http://www.mysql.com/products/workbench/>
- [9] (2014) *Wikipedia: MySQL*. Dostopno na:
<http://sl.wikipedia.org/wiki/MySQL>
- [10] (2014) *E-uspeh: Kaj je MySQL?* Dostopno na:
<http://www.e-uspeh.com/kaj-je-mysql.html>
- [11] (2014) *SERŠ Maribor: MySQL*. Dostopno na:
http://www.s-sers.mb.edus.si/gradiva/w3/omrezja/70_strezniki/mysql.html

- [12] (2014) *Wikipedia: C#*. Dostopno na:
[http://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language))
- [13] (2008) D. Bonačič, *Kratek priročnik jezika C# in razlike z jezikom C++*. Dostopno na: http://osebje.informatika.uni-mb.si/davor/predmeti/orodja_za_razvoj_aplikacij_in_vs3/CSharp/CSharp_zbrano_gradivo.pdf
- [14] (2014) *Wikipedia: .NET Framework*. Dostopno na:
http://en.wikipedia.org/wiki/.NET_Framework
- [15] (2007) B. Črnivec, *Komunikacijske tehnologije*. Dostopno na:
<http://www2.arnes.si/~srcrnivec/KomAsanHP.pdf>
- [16] (2006) M. Munih, *Mikroračunalnik*. Dostopno na:
<http://robo.fe.uni-lj.si/~marko/ur/sinhroni%20in%20asinhroni.pdf>
- [17] (2014) *Wikipedia: RS-232*. Dostopno na:
<http://en.wikipedia.org/wiki/RS-232>
- [18] (2010) B. Pogač, *Telemetrični sistem z mobilnim telefonom*. Dostopno na:
<http://dkum.ukm.si/Dokument.php?id=14090>
- [19] (2014) S. Uranič, *VISUAL C# .NET*. Dostopno na:
<http://uranic.tsckr.si/VISUAL%20C%23/VISUAL%20C%23.pdf>
- [20] (2014) *Serijski vmesnik tehtnice RANGER 3000*. Dostopno na:
<http://dmx.ohaus.com/WorkArea/showcontent.aspx?id=34468>
- [21] (2014) *Primer uporabe*. Dostopno na:
http://osebje.informatika.uni-mb.si/davor/BCKP-OLD/predmeti/programiranje2_vs1/PrimeriUporabeUML/PrimeriUporabe_UML.htm
- [22] (2014) *Podatkovna baza*. Dostopno na:
<https://www.neo-serv.net/gostovanje-podpora/gostovanje/podatkovne-baze-mysql/>
- [23] (2014) *Podatkovni model*. Dostopno na:
<http://drenovec.tsckr.si/model/uvod.htm>
- [24] (2014) *MySQL Connector/NET*. Dostopno na:
<http://dev.mysql.com/doc/connector-net/en/connector-net-introduction.html>

- [25] (2013) *Wikipedia: Uporabniški vmesnik*. Dostopno na:
http://sl.wikipedia.org/wiki/Uporabni%C5%A1ki_vmesnik
- [26] (2014) *Načrtovanje uporabniškega vmesnika*. Dostopno na:
<http://www2.gov.si/mju/emris.nsf/0/C316DA6E36002367C1256EB4007AD1CB?OpenDocument>

